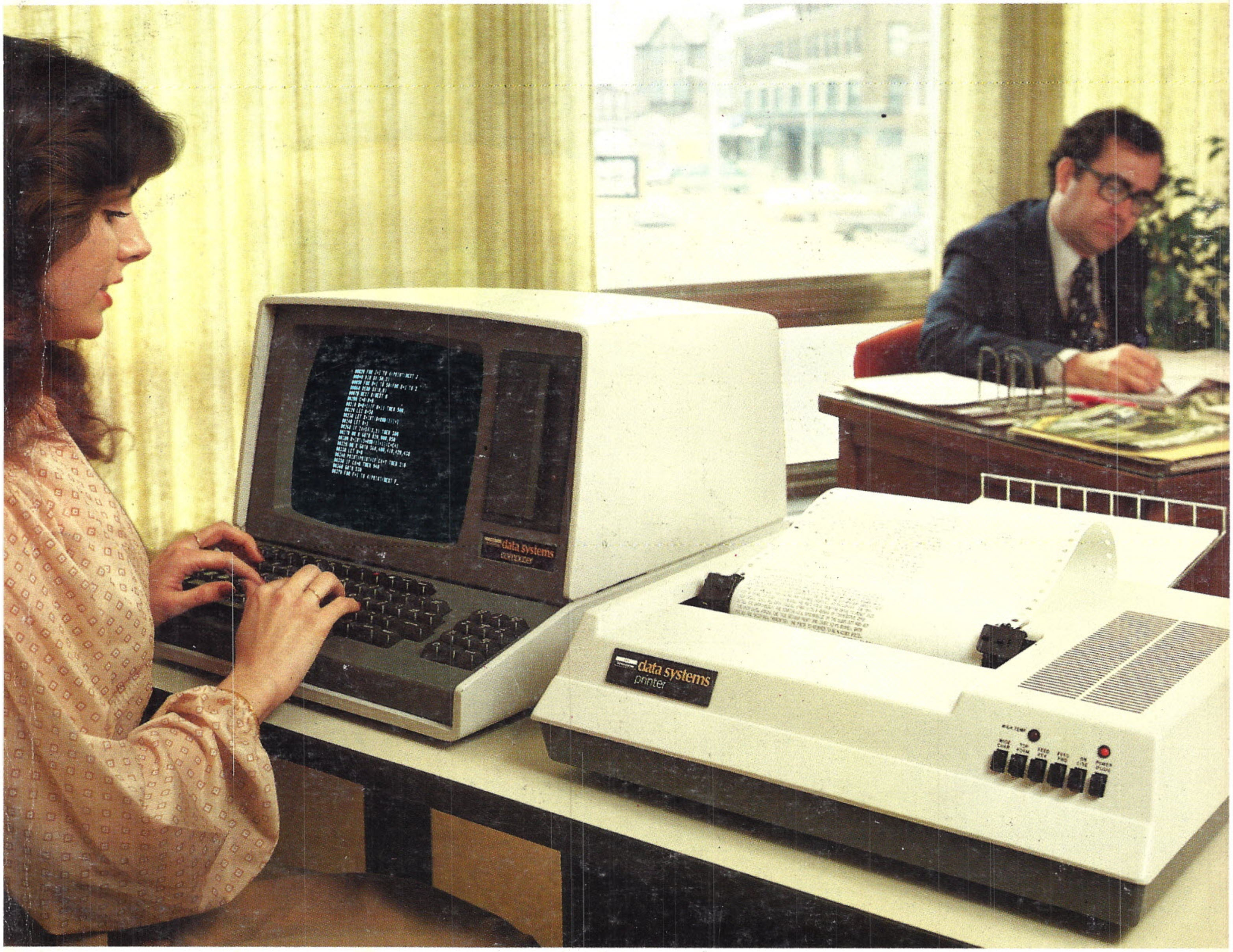
July 1979
$2.50

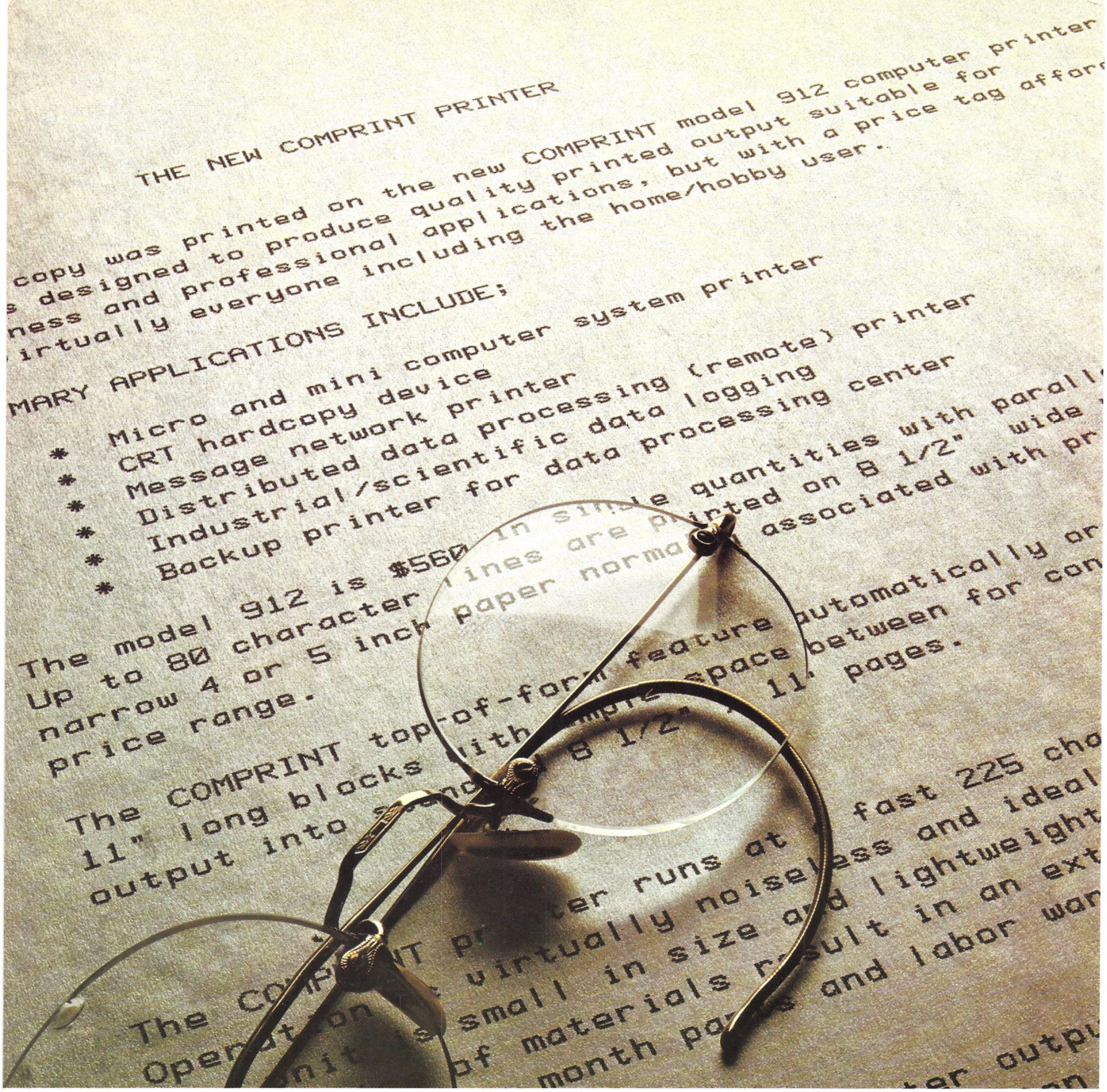
kilobaud

MICROCOMPUTING

T.M.

for business . . . education . . . FUN!




THE NEW COMPRINT PRINTER
copy was printed on the new COMPRINT model 912 computer printer
designed to produce quality printed output suitable for
ness and professional applications, but with a price tag affor
irtually everyone including the home/hobby user.
MARY APPLICATIONS INCLUDE;
* Micro and mini computer system printer
* CRT hardcopy device
* Message network printer
* Distributed data processing (remote) printer
* Industrial/scientific data logging
* Backup printer for data processing center

Reader Service—see page 179

PUBLISHER'S REMARKS

Wayne Green

It's Standard Time

It is actually long past time for the industry to set some standards of compatibility. There was a start back in 1975 with the Kansas City meeting to set cassette standards. The meeting was organized by me when I was still the publisher of *Byte*, and I had the intention of doing my best to make sure that all firms entering the market would adhere to the standard.

OK, here we are four years and a dozen cassette systems later. Let's try to do something about this. I suggest that the first step is to convince those few people who are expert in cassette systems to write some articles pointing out the good and bad points of currently used systems. From there, perhaps we can choose the best of the lot . . . or maybe even come up with a better system as a result of our work over these years. I suspect that the Kansas City "standard" at 1200 baud may turn out to be the best, but let's see some articles on the subject.

That brings us to graphics standards. Every computer system uses a different set of graphics characters. It would be helpful to have a chart showing the various characters sharing the same code and to have an evaluation of the pros and cons of each system. Anyone game to tackle that? An even more challenging test would be a program for converting graphics from one system to other systems.

We have so many disk systems that they are becoming difficult to count: two sizes of disks, hard sectored, soft sectored, single density and double density. It is time for some objective evaluations of these many systems, with an eye toward pinpointing the best of them.

Nothing Can Go Wrong . . . Go Wrong . . . Go Wrong . . .

Joke number 1254C, wherein the captain of the plane gets on the intercom and says, "This is flight 73X, now leaving Boston and flying nonstop to Seattle. This is the first fully automated transcontinental flight run entirely by microcomputer. The system has been thoroughly tested and is so dependable that it is no longer necessary for a flight crew to featherbed on these flights. Be assured that every contingency has been considered and nothing whatever can possibly go wrong . . . go wrong . . . go wrong . . . go wrong . . ."

This brings us to the ridiculous situation we've had trying to keep up with subscriptions. If you get many magazines, you realize that we are not alone in trying to cope with the monumental screw-ups that computers can aggravate. Yes, I know all about computers not making mistakes, but I also know that programmers, computer salesmen, computer manufacturers and data-input people are capable of incredible botching and/or deceptions.

Bill Blair, of *Country Journal*, recently devoted a full-page editorial to apologizing for subscription aggravations to readers, citing Fawcett Publications as the cause of their miseries. Our problems stemmed from Data Input Service Corp (DISC), down near Boston . . . with a big assist from Prime Computer, Inc.

A few years ago, we handled all the subscriptions for *73 Magazine*, our sister publication, by cutting a small mimeograph-type stencil for each subscriber. This paper stencil was then filed in a rack by zip code, and it took a day or two to handle a subscription, complete with sending any desired back issues. When there was any problem, all we had to do was go to the file, pull out the stencil and see what had gone wrong. It took a couple of minutes. One girl was able to handle everything.

Obviously, such an archaic system had to be improved. We called IBM and signed on the dotted line for what they recommended: an IBM 403 with keypunch and card sorter. It was so big and heavy that we had to have a special support in the basement to keep the floor from caving in.

The 403 was no blessing. It took as long to punch a card for each subscriber as it had to cut a stencil . . . and longer to file the cards, which still had to be hand-filed. The 403 printer was not as fast at making labels as the Elliott paper stencil printer. Such was progress.

After evaluating the net result of the new system, which cost about twice as much as the old one to use—and was slower, I started looking for an outside computer service to handle subscriptions. I found one that was recommended highly by one of our advertisers, Waters Manufacturing (it was a subsidiary of Waters, so perhaps their enthusiasm was not without bias), and moved our subscription list there. Within three months we were in deep trouble.

It took us several months to go back over past subscriber lists and find the over 2000 subscribers who had been dropped, seemingly at random, by the computer. This outfit, which eventually became DISC, mumbled about dirt in the tape-read heads, or something.

As the number of subscriptions and the number of problems escalated, with our response time on looking up answers for angry readers going from the one or two days when we were doing everything ourselves to two and three months with the computer service, I was antsy to put things back on home ground. That was when I started calling computer firms to see what I needed, and it was then I found that computer folk had their own language, which I couldn't understand.

My attempt to cope with this language barrier resulted in my starting *Byte*, if you remember that magazine. I eventually gave up trying to learn enough to outwit the computer salesmen and software houses and hired my own "expert" to sort out the claims and promises of the computer firms.

My expert checked out everything available and recommended the Prime 300 computer as being the best for our application. I talked with the Prime salesmen and was assured that their system would be able to handle the subscriptions for *Kilobaud MICROCOMPUTING, 73 Magazine*, our accounting, Reader Service, prospective subscriber lists, industry lists, advertising lists and data, repeater lists, article lists, daily orders, inventory and a few other chores . . . no problem.

By mortgaging everything right down to the paper clips, we were able to buy a Prime 300. Our expert hired two full-time programmers to write programs for handling our subscriptions. One year later, we were ready to give it a try. We were desperate to use it because things had been going from bad to worse with the outside service.

Once we had the subscriptions on our own system, the problems went from worse to total disaster. One 13-megabyte disk was not enough, said Prime, no matter what their salesmen had said. So we bought a second drive. In no time at all, that one had bogged and clogged, and a third drive was needed immediately. I think they ran about $15,000 each, heading us for the poorhouse. When things still grunted to a halt, the diagnosis was a need for 64K additional internal memory . . . another $15,000.

The subscriptions kept the computer from being usable for anything else. Reader Service went back to being done by hand, the accounting had to be sent to a service agency, the payroll was sent to another agency, and so it went. I finally made enough of a fuss that Prime sent a technician to see what

Reader Responsibility

One of your responsibilites, as a reader of *Kilobaud MICROCOMPUTING*, is to aid and abet the increasing of circulation and advertising, both of which will bring you the same benefit: a larger and even better magazine. You can help by encouraging your friends to subscribe to *Kilobaud MICROCOMPUTING*. Remember: Subscriptions are guaranteed—money back if not delighted, so no one can lose. You can also help by tearing out one of the cards just inside the back cover and circling replies you'd like to see: catalogs, spec sheets, etc. Advertisers put a lot of trust in reader requests for information. To make it more worth your while to send in the card, a drawing will be held each month and the winner will get a lifetime subscription to *Kilobaud MICROCOMPUTING*!

This time around, the winner of a lifetime subscription to *Microcomputing* is Reinhard Engel of Los Angeles.

was wrong.

It was then that they discovered that the Prime operating system had no provision for reusing disk space that had been left free by expired or moved subscribers. At this clip, we were heading toward a requirement for hundreds of megabytes. Prime eventually came up with an operating-system mod that allowed material to be deleted from a disk rather than just be ignored.

Added to the system programming problems were troubles with keeping the equipment running. The disks were going down every few days, we had memory problems and the power supply went out every now and then. We finally gave up trying to cope with it and went back to DISC with the subscriptions.

Today, three years into owning the Prime 300, we are able to use it for handling the daily orders, Reader Service and a few shorter lists—and that's about all. Beyond that, it bogs down and little comes out. I estimate that the system is able to do about 20 percent, tops, of what the Prime people promised. I've been trying to get help from Prime on this, but they are so busy building new plants and selling larger systems that they seem unable to remember their sale of three years ago.

I am no fan of IBM. Indeed, when I had IBM come in to recommend a new computer system for us, they suggested a System 32, with floppy disks and no application programming available. I think *that* probably would have been even worse than the Prime. But there is much to be said for buying a more popular computer system and keeping away from smaller outfits such as Prime. It is almost impossible to find programmers or data-processing people with experience on the Prime. This means you have to hire people with little or no training and send them to Prime school at your own expense. Then you have to put them to work for a year or so on a system for which you are paying (they are not cheap) and give them time to gain experience.

The computer has been much more dependable since we fired all of the data-processing people who smoked. They swore up and down that the filters on the system would keep the smoke from hurting the disks, but disk failure now is rare with no smokers around.

For a while it appeared that DISC was going to be able to cope with the subscriptions, but eventually the problems became worse and worse. Three-year subscribers were entered for one year, and address changes brought duplicate, triplicate or no copies. Requests for help went unanswered. It was a disaster. As the complaints grew, I was more and more frustrated. My circulation people assured me daily that everything was being taken care of. All letters were being answered. All missing issues were being sent. Then I found that one of our customer-service people had merely been putting problems into a box and working on the top ones. About 3000 unanswered complaints had built up!

I raised hell. It took about 12 full-time people to work out the problems over a period of months, but it was like building a sand castle with the tide coming in. As fast as 1000 problems were solved, the data-input people at the service bureau would create 2000 more.

We shopped around for a long time, looking for a subscription service (called fulfillment). We checked each one out with several of their customers because many magazines were having serious problems with this, as reported by Bill Blair. The magazine-publishing magazine, *Folio*, often has grim stories of magazine fulfillment service problems; they are a sad fact of modern publishing life.

All this is of little consolation to the innocent subscriber who gets caught in the middle of this mess. Since I subscribe to over 200 publications per month, I frequently run into these frustrations myself. I haven't found any magazines yet that are out to screw anyone. The screwing is there, but it's not intentional. The circulation departments of most magazines are as dedicated to helping the subscriber, and just as frustrated as the subscribers over the problems.

Those readers of *Microcomputing* who have not been loused up by our copelessness are asked to check around to see if they have any friends who have been victimized. Tell them that we really think we have things in hand at last. Our new agency in New York, FAI, seems to be getting good marks from the other magazines they handle and have been squaring away our problems quite satisfactorily for the last two months.

If you have written about a subscription problem and have not heard from us yet, you will receive a customer-service report form in the next two to three weeks . . . this will speed up the handling of all problems and questions. Everyone here wants to have every subscriber happy and fulfilled. Please pass the word around at your club and over the air, and let's make sure that everyone is happy.

Dayton

The Dayton Hamvention, for some reason, has never really pulled in the ham computerists as much as some other shows. For instance, my computer talks at Atlanta usually pull two to three times the audience that turned out at Dayton. Dallas pulled at least double Dayton last year. St. Louis pulled maybe ten times the audience . . . and all are combination ham and computer shows.

Dayton is very well organized and brings in massive numbers of hams. I think the count last year was around 19,000, and this year it was 17,000 or so. That's a mob any way you look at it. It fills the hotels and motels for almost 50 miles around.

Other than Heath, few computer firms were in evidence, again emphasizing the show's absence of computer emphasis. The Dayton convention committee could increase things again if they put some emphasis on microcomputers into their show. I'm sure this would bring out another three to five thousand computer hobbyists and ham computerists who just didn't bother to come this year.

Lose A Few

A kind of simpleminded game program came into Instant Software recently. It would have been instantly rejected except for the graphics, which were outstanding. It was for the TRS-80 and was awfully cute—complete with clever sound effects. Without the graphics, showing three snakes endlessly moving their heads

(continued on page 129)

About the Cover

This month's cover features the new Heathkit® WH89 all-in-one computer and the Heathkit WH14 printer. They are part of a complete new line of personal-computer products from Heath Company. The WH89 has two Z-80 microprocessors, one for the terminal and one for the computer, plus a built-in floppy disk system with a keyboard and numeric keypad.

kilobaud T.M.
MICROCOMPUTING

Publisher/Editor
Wayne Green
Executive Vice President
Sherry Smythe
Assistant Publisher/Editor
Jeffrey D. DeTray
Managing Editor
John Barry
Editorial Assistants
Dennis Brisson
Susan Gross
Administrative Assistant
Dotty Gibson
Production Department
Manager:
Lynn Panciera-Fraser
Staff:
Judith Dalby
Robert Drew
James H. Gray II
Bruce Hedin
Carl Jackson
Dion Owens
Noel R. Self
Robin M. Sloan
John W. White
Typesetting
Barbara J. Latti
Sara Bedell
Mary Kinzel
Holly Walsh
Photography
Bill Heydolph
Tedd Cluff
Projects Editor
Jim Perry
Associate Editors
Phil Feldman
Steve Fuller
Rod Hallen
Len Lindsay
Tom Rugg
Peter Stark
Sherm Wantz
Ken Widelitz
Bookkeeper
Knud E. M. Keller
Marketing/Circulation Mgr.
Robert LaPointe
Bulk Sales Mgr.
Judy Waterman
Asst. Circulation Mgr.
Mary Beth Stanek
Circulation
Pauline Johnstone
Computer Programming
Richard Dykema
Instant Software
Micro Lab
William F. Gollan
Scott Spangenberg
Cida Teixeira
Palmer T. Wolf
Stan Wszola
European Marketing Dir.
Reinhard Nedela
Australian Distributor
Katherine Thirkell
Advertising
Aline Coutu, Mgr.
Bill York
Marcia Stone
Nancy Ciampa
Gaye Halbig
Lori Mugford
Rita Rivard

Kilobaud MICROCOMPUTING (USPS 346-690) is published monthly by 1001001, Inc., Pine St., Peterborough NH 03458. Subscription rates in U.S. and Canada are $18 for one year and $45 for three years. In Europe: Kilobaud erscheint monatlich bei Fachzeitschriftenvertrieb Monika Nedela, 7778 Markdorf, Markstr. 3. Australia: For subscriptions write—Katherine Thirkell, Sontron Instruments, 17 Arawatta St., Carnegie, Vic. 3163 Australia. Please write for other foreign rates. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873. INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION.

OUTPUT FROM ISI

Sherry Smythe

Many letters to Instant Software ask what programs we need. That's kind of like asking a starving man what he wants to order from the menu. Other than games, we're looking for anything. Hundreds of businesses need complete packages of programs written. There are thousands of educational subjects. Games we have.

Perhaps a word of caution is in order: write about something you know. You can get professional advice from people in the field and perhaps put together a good package, but the chances are a lot better if you really know what you are writing about.

Documentation

If you have any questions about what documentation is necessary, you can send for a spec sheet that describes this. You certainly should include information that will help people know what your program does, why they should buy it, what benefits it will have for them, how to use it, any modifications that might be of interest and hints on ways one might get into trouble with it. The material should be typed in uppercase and lowercase, not on a Teletype machine. It should be generously margined for editing.

It is also most helpful if you can include a program listing with comments, a listing of variables and any other pertinent data that will help in evaluation and use of the program. Don't forget to prominently indicate what hardware is required.

We are going to need help with documentation-writing for our finished programs, so if you are interested in joining the associate-editor staff to do writing at home in your spare time, please let us know . . . and give us an idea of your writing background. The value of Instant Software will depend a great deal on the clarity and completeness of the documentation. We want to be sure we have the finest in the business.

Dealers

More and more dealers have been contacting Instant Software, looking for a single, dependable source of programs. With hundreds (or even thousands) of software suppliers coming out of the woodwork, dealers find that they don't have the time to send for programs; evaluate them; order one or two from each source; contend with lousy documentation, unloadable cassettes, poor packaging, an infinite amount of bookkeeping, a wide variety of prices and a lot of duplication of programs. They also are growing increasingly concerned over the liability when programs of questionable parentage are concerned.

It is going to become more difficult for small software firms to sell to stores. At Trenton we were approached by several people who had tried to wholesale their programs and decided it was time to let Instant Software do the marketing.

Openings

Despite continued hiring of more people, we have many more spots for people who are interested in developing careers in microcomputing and do not smoke. We need a good microcomputer technical editor, assistant editors, a strong FORTRAN programmer, a DP manager, writers, BASIC programmers, a microcomputer technician, plus marketing and production help.

PET-POURRI

Len Lindsay

The New PET Arrives

The new 32K PETs with full-size keyboards really exist! I have had one for two days now and can give you a preliminary report.

The 32K PET costs $1295 (price was raised $100). It has the same number of keys as the 8K PET (plus a shift-lock key) but in a typewriter-style keyboard. The built-in cassette unit had to be removed to accommodate the larger keyboard. The Commodore external cassette unit #1 ($95) plugs into the port in the rear of the new PETs. Unit #2 is the now-empty plug inside the PET. The keytops are marked boldly with the characters. Graphic symbols are etched on the front side of the keys and look very nice. There is a long space bar, and return, shift, shift-lock and reverse keys are oversized. The numeric keypad was retained with the same key layout.

Most of the bugs in the BASIC and operating system are corrected with the new ROMs that come with the new PETs. Unfortunately, these new ROMs use a different set of locations in PET's memory in pages 0, 1 and 2. PEEK and POKE locations are in different areas, and, thus, programs using them will probably have to be modified to run correctly on the new PETs.

To help new PET users modify old PET programs, I wrote a program called POKE/PEEK FIND & LIST (see program listing). It will search a PET BASIC program and list the line numbers that contain PEEKs or POKEs, along with the location affected. I also have compiled a preliminary

```
63900 CLR:REM POKE/PEEK FIND&LIST
63901 X=1024:Y=0:DIM Z1(255),Z2(255),PV$(255),LC(255)
63902 PRINT "[CLR]RESULTS TO [RVS]S[OFF]CREEN OR [RVS]P[OFF]RINTER?"
63903 FOR CB=1 TO 10:GET CB$:NEXT CB:REM CLEAR KEYBOARD BUFFER
63904 GET A$:IF A$="" THEN 63904
63905 IF A$<>"P" THEN GOTO 63907
63906 PRINT:INPUT "DEVICE NUMBER  5[ 3 LEFT]";DV
63907 A=PEEK(X):REM VALUE IN LOCATION X
63908 IF A=0 THEN LC=LC+1:PRINT LC;:Z2=Z1: Z1=X:REM LAST TWO DELIMITERS
63909 IF Z2+1=Z1 AND PEEK(X+1)=0 THEN EC=Y: GOTO 63931: REM END OF PROGRAM
63910 IF A=151 OR A=194 THEN GOSUB 63914:REM 151 IS VALUE OF POKE 194 IS PEEK
63911 X=X+1
63912 IF X>9000 THEN EC=Y: GOTO 63931: REM SAFETY CHECK FOR END OF PROGRAM
63913 GOTO 63907:REM NEXT LOCATION
63914 P=X:REM POKE/PEEK LOCATION
63915 PRINT "[RVS]"Y+1"[LEFT] [OFF]";:REM PRINT POKE COUNT
63916 C=0:REM DIGIT COUNTER
63917 Y=Y+1:REM POKE COUNTER
63918 LC(Y)=Z1:REM LOCATION OF DELIMITER
63919 PV$(Y)="":REM INITIALIZE DIGIT STRING
63920 Z1(Y)=(256*(PEEK(Z1+4)))+(PEEK(Z1+3)):REM LINE NUMBER THIS LINE
63921 Z2(Y)=(256*(PEEK(Z2+4)))+(PEEK(Z2+3)):REM LINE NUMBER LAST LINE
63922 P=P+1:REM NEXT LOCATION TO CHECK
63923 PV=PEEK(P):REM VALUE IN LOCATION
63924 IF PV=32 OR PV=96 OR PV=41 THEN 63922:REM SPACE, SHIFT SPACE, "("  -SKIP
63925 IF PV=44 THEN RETURN:REM COMMA THUS DONE
63926 IF PV<48 OR PV>57 THEN PV$(Y)=PV$(Y)+"?":RETURN:REM NONDIGIT BEFORE COMMA
63927 PV$(Y)=PV$(Y)+CHR$(PV):REM ADD NEXT DIGIT TO STRING
63928 C=C+1:REM DIGIT COUNTER
63929 IF C>5 THEN PV$(Y)=PV$(Y)+"?":RETURN:REM TOO MANY DIGITS
63930 GOTO 63922
63931 IF DV THEN OPEN 5,DV : CMD 5 :REM OPEN PRINTER
63932 REM *** PRINT POKE/PEEK LINES NOW ***
63934 PRINT:PRINT "PROGRAM CONTAINS"LC"OR LESS LINES":PRINT "AND"Y"POKES/PEEKS"
63935 PRINT "[RVS] LINE#   POKE#  DLM LOC  PREV LINE"
63936 FOR Y=1 TO EC
63937 Q=Z1(Y):GOSUB 63953:PRINT Q;
63938 PRINT " ";
63939 Q$=PV$(Y)
63940 FOR Q=1 TO 8-LEN(Q$):PRINT " ";:NEXT Q
63941 PRINT Q$;
63942 Q=LC(Y):GOSUB 63953:PRINT Q;
63943 PRINT "  ";
63944 Q=Z2(Y):GOSUB 63953:PRINT Q
63945 IF INT(Y/20)<>Y/20 THEN NEXT Y : REM PRINT 20 LINES THEN GO TO NEXT LINE
63946 FOR CB=1 TO 10:GET CB$:NEXT CB:REM CLEAR KEYBOARD BUFFER
63947 GET CB$:IF CB$="" THEN 63947
63948 IF DV THEN CMD 5 : REM KEEP PRINTER ON LINE
63949 PRINT "[RVS] LINE#   POKE#  DLM LOC  PREV LINE"
63950 IF EC-Y+1 THEN NEXT Y
63951 IF DV THEN PRINT#5,"":CLOSE 5:REM TURN PRINTER OFF LINE
63952 END
63953 IF Q<10 THEN PRINT " ";
63954 IF Q<100 THEN PRINT " ";
63955 IF Q<1000 THEN PRINT " ";
63956 IF Q<10000 THEN PRINT " ";
63957 RETURN
```

PEEK/POKE FIND & LIST.

chart that gives the new location for many of the old ones. (See "Teach an Old PET New Tricks," p. 72.) Please contact me if you have a better chart or program for modifying old PET programs.

To illustrate how these new POKE locations change, a few examples are given below.

Clear keyboard buffer and get a character

Old PET

```
POKE 525,0:WAIT 525,1:GET A$
```

New PET

```
POKE 158,0:WAIT 158,1:GET A$
```

Both PETs accept the following:

```
100 FOR X=1 TO 10:GET A$:NEXT X:REM clear buffer
110 GET A$:IF A$="" THEN 110
```

Keyboard buffer

The keyboard buffer for the old PET was from 527-536. The character count was kept in 525. The keyboard buffer for the new PET is from 623-632. The character count is kept in 158.

Top of memory pointer

Old PET

```
PRINT 256*PEEK(133)+PEEK(132)
```

New PET

```
PRINT 256*PEEK(51)+PEEK(50)
```

Stop key disable

Old PET

```
POKE 537,136 or POKE 537,PEEK(537)+3
```

New PET

```
POKE 144,46 or POKE 144,PEEK(144)+3
```

Smart Program

In the old PET, the BASIC in ROM was protected. Any time you PEEKed at any of these locations, PET would return a value of 0 (zero). The new PET is not protected in this manner. Thus a simple PEEK to a location such as 51234 will allow your program to determine which PET it is running in. With this information, you can write programs that use PEEK and POKE according to the type of PET. These will be smart programs. This calls for an example.

The first line in your PET program could be:

```
1 PT=PEEK(51234):IF PT THEN PT=1
```

The above line will set PT=0 (PT stands for PET Type) if it is an old PET, or set PT=1 if it is a new PET.

Program lines will now be longer and more complex, but will work in both PET varieties. Use the correct POKE or PEEK location in the old PET and add or subtract an amount that would bring the location to that of the new PET; make sure to multiply by PT. On the new PET, PT=1 and the number will be added or subtracted as noted. On the old PET, PT=0, and 0 times anything is 0; thus 0 will be added or subtracted, leaving the original amount.

Now a simple way to disable the STOP key on either old or new PETs is:

```
1 PT=PEEK(51234):IF PT THEN PT=1
2 POKE 537-PT*393,PEEK(537-PT*393)+3
```

Next we can clear the keyboard buffer and get an answer to our question in the following manner:

```
100 PRINT"Need instructions?"
110 POKE 525-PT*367,0 : WAIT 525-PT*367,1 : GET A$
120 IF A$ = "Y" THEN GOSUB 5000 : REM Print instructions
130 Program continues here. . .
```

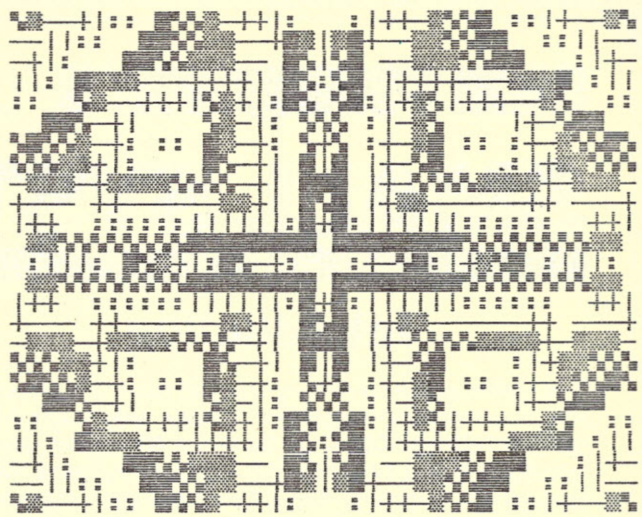
There are more differences between the old and new PETs. The memory expansion connector no longer is an extension of the PC board. It now is inside the PET as long spikes pointing up, similar to the PET keyboard connector. The main board is laid out quite differently too. The new PETs still have either graphics or lowercase mode, but now the lowercase is unshifted. You must shift for uppercase, just as on a typewriter. When I print it on my Teletype 43 via Connecticut Microcomputer's RS-232 adapter, it prints reversed—lowercase is printed as upper. It is unfortunate that Commodore has once again made things hard for their PET users.

Commodore News

As noted above, the new 32K PET price went up $100 to $1295. The PET 2021 electrostatic printer was canceled and will not be produced. The PET 2040 dual floppy-disk unit went up $200 in price to $1295. The PET 2041 single-disk unit has been canceled due to lack of interest in it. It is being replaced by the 2040A single floppy for $895. This is the dual floppy unit with only a single drive installed. I should have a dual floppy-disk unit by next week and will give a full report on it next month.

Program Protection Revisited

Last month I told you about BC Computing (2124 Colorado Ave., Sun Prairie WI 53590), which has the best protection method for the 8K PET. Since Commodore changed things in the new PET, programs protected on the old PET will not be protected on the new PET (and vice versa). But BC Computing is already in the final stages of developing a protection method that will work with either PET. So all is not lost due to Commodore and the new locations in memory.



Example of Kaleidoscope program.

Rumors

The latest rumor I have confirmed is that the PET will soon be called the CBM. Will I have to change my *Microcomputing* column title from PET-pourri to CBM-pourri? That doesn't sound right!

Full Graphics Printer

I was fortunate to receive an Axiom (5932 San Fernando Rd., Glendale CA 91202) printer. Axiom has two models that simply plug into the PET. Both are electrostatic and both print all PET graphic characters. The Axiom PET Printer costs $535 and is back-ordered about 16 weeks. The Axiom PET Plotter costs $895 and is more readily available. Both print on a roll of 5-inch-wide special "silver" paper. The Plotter can print data, reports, list programs or print an exact copy of what is on the PET screen (screen dump program included on tape with the unit). It has two letter sizes, software selectable—either 40 or 80 characters per line. It also has two modes, graphics or lowercase, as well as a reverse feature, all software controlled. The printer has all the functions of the plotter, but will not dump screen contents without leaving a thin blank line between each line from the screen. The print copies very well. Please note the example of the Kaleidoscope program printed with the Axiom Plotter.

Cassettes

Important for all PET users are cassettes and how to store them. Remember, you are dealing with a picky computer, not a stereo playing a tape of the Fugs. Choose cassettes for use with your PET with extreme care. If anything goes wrong with your cassette you lose hours, days and even weeks of work. To prevent this loss, use only the very *best* cassettes. To be extra safe, keep two copies of each program—on two different tapes—stored in two different places. The safest place to store your cassettes is in a metal box or cabinet away from heat and high humidity. Since this is not practical, at some times, for tapes used daily, it may be a good idea to keep "master" tapes of your programs and store these in a metal box. These masters can be C-30 tapes with many programs on each side. Fast access is not important because you should have no need to use the master tape.

I recommend using only C-10 cassettes for regular program storage. Programs can be stored on the front and back, and each will be easily and quickly accessible. I have used most of the tapes you see advertised and had *many* problems. Some tapes were no good to start with. I couldn't even find programs saved on Scotch tapes. Microsette seemed very

(continued on page 129)

OHIO SCIENTIFIC'S ✓013
SMALL SYSTEMS JOURNAL

Introduction

This month we are continuing with part two of our discussion about information-management systems specifically dealing with OS-DMS and OS-MDMS for 8-inch Winchester Disk and mini-floppy-based Ohio Scientific computer systems. We will continue the discussion with a formal definition of terms and then present some operational examples of OS-DMS.

OS-DMS DEFINITION OF TERMS

Introduction

Concept of DMS

Standard ways to manage and manipulate data.

Standardization.

Makes operations simpler.

Programming quicker since you're not reinventing the wheel.

Utilities provide performance not practically attainable on stand-alone programs.

History of DMS

DMS was first introduced for large-computer installations eight years ago. It recently became very popular as large computers became more accessible via terminals. According to a recent large-computer report, DMS or DBMS systems are in use at about 4000 "mainframe" installations.

OS-DMS Dictionary

DMS uses the following structures:

File: A collection of information in an organized fashion. In OS-DMS, files are always on floppy disk or hard disk. (In OS-DMS, files can be of arbitrary length up to 74 million bytes.)

Record: A subdivision of a file that contains a collection of related information. (The information for a single mailing label corresponds to one record in a mailing-label file.)

Fields: The records are further subdivided to cells that store one parameter called fields. (The zip code, city and state are all examples of possible fields.) In OS-DMS, fields can be up to 72 characters.

Entry: An entry is the physical information in a specific field. In OS-DMS, entries are typically smaller than their corresponding fields. The extra character positions are normally nulled. (The number 44202 is the entry in the zip-code field of one record in one file.)

Editor: To edit a file is to change entries in a file. Editing includes placing initial entries in a file. An editor is a special program that guides the user through file changes.

Label: In OS-DMS, each field must have a descriptive label. The first field of each record of the file shares a unique label, etc. (The field of a mailing-label file that stores the zip code could have the label "Zip.")

Key: A key is a field within a record or a special file that has special significance in obtaining an answer to or result of a specific problem. Under OS-DMS, keys can be any field within a record, or in special files that point to records in other files. These special files are called key files. (To obtain the address of an individual, one would check the "NAME" field in the mailing-list file. Thus, for this operation, the "NAME" field is the key field. Frequently, one wishes to generate mailing lists in zip-code order. For this application, it may be desirable to create a special "KEY" file containing the zip-code fields and order (sort) them in some specific order. This key file then pulls records from the "Master" file to print labels.)

Index: The index is the virtual field address of an entry field, record or file. Ultimately, the system must construct the index before obtaining information from a file. In OS-DMS, indexes can range from 0 to 99 million. The term index is analogous to the reserved-word index in OS-65U's BASIC. OS-65U's virtual data memory architecture via the continuous mapping function "INDEX" allows OS-DMS to easily perform functions never achieved on other DMS systems, regardless of computer size or system cost.

Record Number: The record number refers simply to the order of the record in the file. Thus, the fourth record in a file would have record number "4." Record numbers are only useful in fixed record-length files such as OS-DMS "Master" files. Record numbers provide fast access to records, but are of little use to humans because of their lack of association with the file contents. (It is much easier to remember a person's name than the record number of the mailing address.)

Field Number: The field number simply refers to the order of a particular field within a record. The discussion for record number applies.

Dictionary: A special file and program system that maintains explanations of contents and intended purpose of master files and their associated key files. This information is usually stored in the form of comment strings and in system-understandable form so that the system "knows" that the file MLABEL (for instance) is a file containing the names and addresses of this month's sales leads.

Query: A program system that assists an untrained user in obtaining answers to specific questions from the available data base. In early DMS versions, Query simply presents all possible options to the user via a logical tree structure. In advanced versions, Query accepts simple English sentences and parses them using techniques similar to the popular "Elisa" program. Typical uses for Query are finding an address or phone number or performing a stock check. News-media demonstrations of Query systems have promoted the consumer concept of "asking the computer" or the "computer says. . ." The Query system must have a good dictionary and accurate field labels for successful operation.

File Access: Accessing a file is the act of putting information in or taking it out. Mechanically, files can sometimes be subdivided into sequential and random-access types. "Conventional" sequential files have entries immediately following each other. They usually do not have discernible record or field structures. "Conventional" random-access files organize information into records that are usually of fixed length determined by some equipment limitation such as disk sector length. File structures under OS-65U, and hence OS-DMS, are not subject to these traditional or conventional rules or limitations. All OS-65U files are randomly ad-

dressable and are usually sequentially addressable (without undesirable effects). The only real distinction is whether entries immediately follow each other or are placed at predescribed indexes sometimes leaving gaps in the files. Under OS-65U, the first type is referred to as a sequential file, and the second is called a random file, simply for lack of a better name. OS-65U random files are easier to "Edit" or change. However, sometimes the sequential-file type yields faster access.

In real-world applications, one desires to extract information from very large multipurpose files very quickly. This usually requires the use of one or more levels of "Key" or "Index" files that act as a table of contents or directory to the master file. This technique is referred to as the Index Sequential Access Method, or ISAM, file system. OS-DMS directly supports ISAM files with up to eight key files per master file. The system also supports more complex techniques such as cross-indexing and multi-level indexing in conjunction with user programs. However, such techniques should be left to those who completely understand them because they do not necessarily improve performance. It is more suitable to discuss file access from an end-user viewpoint than to throw around terms such as random, sequential or ISAM.

OS-DMS supports three procedures for file access:

A. Numeric
B. Associative
C. Conversational

Numeric Access: Makes use of record numbers, field numbers or pure indexes to access files. This approach yields the fastest access, but is impractical for users and general-purpose programs. Problems occur when a file is reordered or edited in some other fashion.

Associative Access: Uses labels to identify fields that are accessed and compared to the test string. For example, a program could request the contents of the field labels "Phone Number" in the record that has a field labeled "Name" containing the string "John Doe."

This program does not need to know the file structure of the record so that it can operate on any OS-DMS-generated file that has name and phone-number fields. Associative Access techniques are currently in the forefront of software technology. OS-DMS is specifically designed to provide high performance, yet simple-to-use Associative Access capability.

Conversational Access: A technique (very much in its infancy) that allows information retrieval without knowledge of the file system or computing in general. Query systems and some advanced report-writers approach this ability. It is strictly for users to obtain information (usually very slowly) without programming.

Advanced versions of OS-DMS will incorporate increasingly refined Conversational Access abilities.

Using OS-DMS

There are three fundamental steps in the use of the information-management system. The first step is to create the file containing the information desired, the second step is editing the information and the third step is outputting the information selectively as desired. Hence, all information-management-related tasks can be categorized under three categories: entry, editing and outputting. Let's now examine the most rudimentary operations in each of these categories of OS-DMS to gain some insight as to how the system works.

Entry

The first step to utilizing the information-management system is for the user to organize the information in logical fashion selecting the format of records, including what will be stored in pertinent fields, how long these fields will be and defining the labels of these fields. Under OS-DMS, the file creation and loading is handled under one program. When the system is first brought up under OS-DMS, the following menu appears.

```
OS-DMS  2/79

FUNCTIONS
---------
(1)  CREATE NEW MASTER FILE
(2)  CREATE NEW KEY FILE
(3)  EDIT MASTER FILE
(4)  LOAD KEY FILE FROM MASTER
(5)  EDIT KEY FILE
(6)  DUMP MASTER FILE
(7)  GENERATE MAILING LABELS FROM MASTER FILE
(8)  MASTER FILE MERGE OR LOAD
(9)  DISKETTE COPIER
(10) MULTI-FILE MULTI-FORMAT REPORT WRITER
(11) MULTI-CONDITIONAL REPORT WRITER WITH STATISTICAL FUNCTIONS
(12) MULTI-CONDITIONAL STATISTICAL PACKAGE
(13) SORT A FILE
(14) MASTER FILE RECORD INSERTER
(15) MASTER FILE RECORD DELETE & REPACK
(98) DMS FILE DIRECTORY
(99) EXIT

ENTER THE NUMBER OF THE FUNCTION DESIRED ? 99

PASSWORD? PASS
```

To initially enter information, selection 1 is specified. Diagram 2 indicates the dialogue for the creation of a hypothetical automobile-inventory file. The user has specified a very short file of ten records, each containing seven fields. The fields are make, year, model, miles, engine, condition and price, with a user-specified length in each field.

```
OS-DMS CREATE NEW MASTER FILE
-----------------------------
DEVICE TO STORE MASTER FILE ? A
FILE NAME, PASSWORD ? AUTOS,PASS
TOTAL NUMBER OF RECORDS IN MASTER FILE ? 10
NUMBER OF FIELDS PER RECORD ? 7

FIELD 1 LABEL ? MAKE
        LENGTH ? 8

FIELD 2 LABEL ? YEAR
        LENGTH ? 4

FIELD 3 LABEL ? MODEL
        LENGTH ? 8

FIELD 4 LABEL ? MILES
        LENGTH ? 6

FIELD 5 LABEL ? ENGINE
        LENGTH ? 2

FIELD 6 LABEL ? CONDITION
        LENGTH ? 6

FIELD 7 LABEL ? PRICE
        LENGTH ? 4

MASTER FILE NAME            AUTOS0
LOCATED ON DEVICE           A
FILE TYPE                   DATA
ACCESS RIGHTS               READ
PASSWORD                    PASS

FIELD                           LENGTH
*****                           ******

MAKE                              8
YEAR                              4
MODEL                             8
MILES                             6
ENGINE                            2
CONDITION                         6
PRICE                             4


TOTAL NUMBER OF RECORDS  10
TOTAL FILE LENGTH  561

     IS THIS CORRECT (Y OR N) ? Y


EXIT (E) OR CONTINUE (C) ? C
```

```
OS-DMS MASTER FILE LISTER
-------------------------

CONSOLE (C) OR PRINTER (P) OUTPUT ? P

DEVICE WHERE MASTER IS STORED ? A
MASTER FILE NAME, PASSWORD ? AUTOS, PASS

FILE: AUTOS
NUMBER OF RECORDS:  15

 MAKE                                --------
 YEAR                                ----
 MODEL                               --------
 MILES                               ------
 ENGINE                              --
 CONDITION                           ------
 PRICE                               ----

RECORD:  1                          INDEX:  112
 MAKE                                FORD
 YEAR                                1977
 MODEL                               SEDAN
 MILES                               23745
 ENGINE                              V8
 CONDITION                           GOOD
 PRICE                               4590

RECORD:  2                          INDEX:  157
 MAKE                                CHEV
 YEAR                                1975
 MODEL                               2 DR
 MILES                               38987
 ENGINE                              6
 CONDITION                           VRGOOD
 PRICE                               3150

RECORD:  3                          INDEX:  202
 MAKE                                OLDS
 YEAR                                1976
 MODEL                               4 DR
 MILES                               42025
 ENGINE                              V8
 CONDITION                           EXCEL
 PRICE                               3999
```

Skipping ahead slightly, Diagram 3 lists the contents of the first few records of the file after it has been edited. The master file lister used here includes diagnostic information such as the absolute record number and the OS-65U file index for the beginning of each record. The important features to recognize at this stage are that the user specifies the syntax of the file labels that are permanently stored in its header. Also, records, fields and files can be of arbitrary length and are not limited in any way by artificial boundaries such as sectors or tracks. Under OS-65U, an individual DMS file could be up to 74 million bytes long, with each record containing hundreds of fields.

Editing

The basic editing tool under OS-DMS is the edit master file function selection 3. In the following examples, a real-estate listing file is utilized. Once the selection is made, the editor signs on and asks for the file name and password. When passwords are not used, a period is used as a default.

```
OS-DMS EDIT MASTER FILE
-----------------------
DEVICE MASTER FILE STORED ON ? B
FILE NAME, PASSWORD ? REALI,

DO YOU WANT TO SKIP THE INSTRUCTIONS ?
```

The editor then offers the user a wide range of accessing techniques. These techniques allow the user to access the information for examination and/or modification.

```
RECORD SELECTION TECHNIQUES

(#) RECORD NUMBER ((A) FOR APPEND)
(L)  LABEL/CONTENTS SEARCH
(K)  KEYFILE CONTENTS SEARCH
(S)  SEARCH MASTER FOR MATCH
(/)  EXIT                        ? L
```

Briefly, the options are specified in absolute record number with a pound sign (#) followed by a record number or simply an "A" to append to the end of the file. L specifies label-contents search in which the user specifies the label and the desired contents a file is searched for. Key-file-contents search allows the user to scan one of eight ISAM key files for a match. This technique is particularly well suited for large files over a few hundred thousand bytes in length. Finally, the absolute search allows the user to enter up to a 32-character string in which the file will be searched for an absolute match. In this example, the user specifies L for label-of-content search. The system then presents the user with all possible labels present in the file he is operating on as shown in Diagram 6.

```
FIELD LABELS

 1  OWNER                                  2  STREET
 3  CITY                                   4  ROOMS
 5  BEDROOMS                               6  PRICE

ENTER  FIELD LABEL OR ↑#, CONTENTS

   ? ↑1, SUMMERS
```

The user then can type in the name of the label or the label number preceded by an up-arrow and the contents he desires. In this case he has entered OWNER SUMMERS, hence, the operating system will proceed to scan the real-estate file for a record that has an OWNER = SUMMERS. Diagram 7 shows the results. Record 7 contained OWNER SUMMERS. The operating system then asks the user if this is the correct record; if he says "no" it continues to search to end of file. The user may have utilized the editor to simply obtain information about the house that was owned by Summers or he may have wished to modify the information file. In this case the system asks for a field label or number that the user wishes to edit. Here the user specified 3, which corresponds to the field CITY. The dialogue is completed in the example by presenting the field CITY to the user so he can modify it if desired. Note that there are two levels of password security in the system with a password on the editor and a password on the file. This safeguards the system against unauthorized tampering or editing of the master fields.

Output

There are several output forms in a highly developed information-management system. However, possibly the most useful and popular output form is the universal report-writer. The following example is an extremely simple utilization of Ohio Scientific's multi-conditional report-writer. The dialogue is quite lengthy but can be recorded for future use under a one-word command by use of Ohio Scientific's optional Query System, so that once a dialogue for a particular report is specified, it can be called up by the user with a single-word command.

```
RECORD:  7

FIELD                                CONTENTS
 1 OWNER                              SUMMERS
 2 STREET                             98 THOMPSON
 3 CITY                               AKRON
 4 ROOMS                              9
 5 BEDROOMS                           4
 6 PRICE                              81500
IS THIS THE CORRECT RECORD ? Y
FIELD LABEL OR NUMBER ? 3

FIELD: CITY
     AKRON
     ---------------
```

```
CONSOLE OR PRINTER ? P

SINGLE OR DOUBLE SPACED REPORT ? S

VERTICAL OR HORIZONTAL FORMAT (V OR H) ? H

OS-DMS MULTI-CONDITIONAL REPORT WRITER
----------------------------------------


KEY FILE OR MASTER FILE ACCESS ? M

DEVICE MASTER FILE IS STORED ON ? B

MASTER FILE NAME,PASSWORD ? AUTOS
?? PASS

PLEASE WAIT


ENTER REPORT HEADING ? CURRENT FORDS IN STOCK

******************************************************************
*  ALL FIELD NAMES WILL BE DISPLAYED WITH THEIR FIELD NUMBER     *
*  SPECIFY IF A PARTICULAR FIELD SHOULD APPEAR ON THE REPORT     *
*  BY ENTERING A '↑'FIELD NUMBER FOR EACH FIELD  IF YOU WANT     *
*  ALL THE FIELDS IN A FILE PRINTED ON THE REPORT IN THE SAME    *
*  ORDER THEY APPEAR IN THE FILE, TYPE THE WORD 'ALL' INSTEAD    *
*  OF '↑'FIELD NUMBER. <<NOTICE>>  ENTER THE FIELD NUMBERS IN    *
*  THE ORDER YOU WISH THE FIELDS TO BE PRINTED ON THE REPORT!    *
******************************************************************

(1)  MAKE                            (2)  YEAR
(3)  MODEL                           (4)  MILES
(5)  ENGINE                          (6)  CONDITION
(7)  PRICE


? ↑1↑3↑2

DO YOU WISH TO CHANGE ANY HEADINGS (Y OR N) ? Y

DO YOU WISH TO CHANGE HEADING #  1   :MAKE ? Y
ENTER NEW HEADING ? CAR MAKE

DO YOU WISH TO CHANGE HEADING #  2   :MODEL ? N

DO YOU WISH TO CHANGE HEADING #  3   :YEAR ? N


HOW MANY CONDITIONS DO YOU WISH TO SET  (0>4) ? 1

(1)  MAKE                            (2)  YEAR
(3)  MODEL                           (4)  MILES
(5)  ENGINE                          (6)  CONDITION
(7)  PRICE
```

```
'↑'FIELD # OR LABEL TO KEY ON FOR CONDITION # 1 ? ↑1

MAKE VS CONSTANT (1)
MAKE VS CONTENTS (2) ? 1

IF CONDITIONS POSSIBLE (3-7 FOR NUMERIC VARIABLES ONLY)

(1)   MAKE IS EQUAL TO
(2)   MAKE IS NOT EQUAL TO
(3)   MAKE IS LESS THAN
(4)   MAKE IS LESS THAN OR EQUAL TO
(5)   MAKE IS GREATER THAN
(6)   MAKE IS GREATER THAN OR EQUAL TO
(7)   MAKE IS WITHIN A SPECIFIC RANGE ? 1

CONSTANT MAKE IS TO BE COMPARED AGAINST ? FORD

DO YOU WISH TO GENERATE ANY STATISTICAL INFORMATION ? Y

ENTER THE SELECTIONS WHICH YOU WOULD LIKE TO SUM
IF NONE, SIMPLY HIT THE RETURN KEY


(1)   CAR MAKE                          (2)   MODEL
(3)   YEAR
ENTER YOUR SELECTION(S)
?

ENTER THE SELECTIONS YOU WOULD LIKE TO HAVE AVERAGED
IF NONE, SIMPLY HIT THE RETURN KEY


(1)   CAR MAKE                          (2)   MODEL
(3)   YEAR
ENTER YOUR SELECTION(S)
? ↑3

YOU MAY NOW SELECT ANY TWO FIELDS. THESE TWO FIELDS
WILL THEN BE MULTIPLIED TOGETHER AND THE SUM OF THEIR
PRODUCTS WILL BE DISPLAYED AT THE END OF THIS REPORT
IF YOU DO NOT DESIRE THIS OPTION SIMPLY HIT THE RETURN KEY

(1)   CAR MAKE                          (2)   MODEL
(3)   YEAR
ENTER YOUR SELECTION(S)
?
```

```
CURRENT FORDS IN STOCK
--------------------------------------------------------------
--------------------------------------------------------------

CAR MAKE     MODEL        YEAR
----------------------------

FORD         SEDAN        1977
FORD         WAGON        1974


CURRENT FORDS IN STOCK
--------------------------------------------------------------
--------------------------------------------------------------

                                                      AVERAGES
--------------------------------------------------------------
YEAR
----
 1975.5
--------------------------------------------------------------
```

In the next issue of the journal, we will continue our exploration of features of OS-DMS.

BOOKS

Programming a Microcomputer: 6502
Caxton C. Foster
Addison-Wesley, Reading MA
1978, Softcover, 240 pages, $8.95
vs
Programming the 6502
Rodney Zaks, Sybex, Inc.
Berkeley CA, 1978
Softcover, 305 pages, $9.95

So you have a KIM, SYM, VIM, AIM, PET, Apple, Jolt or some other 6502-based microcomputer system (perhaps even your own "home brew" version). You've passed the acquisition hurdle, and now you are looking for a good book to help you program in assembly language. Then either of these two books will do the job, but since they are written in totally different styles, perhaps one or the other would be better for you.

Although both books were copyrighted in 1978, Foster's is actually older (based on its Library of Congress number), so let's consider it first.

Excluding the first two chapters ("Basic Ideas" and "Console Controls"), essentially the Foster book contains 12 problems for the reader to solve using 6502 assembly language. Each chapter introduces a new problem and describes it in some detail, develops one or more flowcharts, discusses any problems likely to be encountered and leads the reader toward the final solution step by step. (One of the appendices even gives a solution for those who have trouble developing their own.)

The problems are interesting and cover a wide range of applications from the simple ("Morse Code Oscillator"—Chapter 3) to the sublime (an interpreter and an assembler for a "Dream Machine"—Chapters 13 and 14). The other chapter problems are: (4) "Piano Keyboard," (5) "Keybounce," (6) "Combination Lock," (7) "Tune Player," (8) "Digital Clock," (9) "Tracking (analog and digital)," (10) "Microbart," (11) "Top Secret and Discrete" and (12) "Elevator Control." Each new problem generally provides the "excuse" to introduce another 6502 hardware or programming feature (such as zero-page addressing, input/output) and the net result is a lot of information attractively packaged into several little problems.

However, according to the introduction to Chapter 1, "In addition to this book, you will need a . . . 'hardware' manual for the machine you have. . ." In other words, this book is not complete in and of itself. Specifically, although the first two chapters do appear to begin at the beginning, and although two of the five appendices list the 6502 instructions and addressing modes and describe them briefly (using only eight pages!), there really is no attempt to teach *systematically* how to program the 6502.

Apparently it is the author's hope that readers who already know a little about the 6502 will learn how to program it better using his "problem-solving" approach. From my own experience (teaching beginning assembly-language programming to college students), a "hands-on" approach is good only if preceded by a strong and systematic undergirding of knowledge about the machine's characteristics; this foundation is missing in Foster's book.

On the other hand, Zaks' book, as the preface points out, "has been designed as a complete self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should also be of value to anyone using the 6502." It does seem to be complete, for not only does it systematically lay a strong foundation of knowledge about programming in general as well as coding for the 6502, it also contains sufficient material and is well enough organized for use as a reference text after the initial reading.

This book contains eleven chapters: Basic Concepts, 6502 Hardware Organizations, Basic Programming Techniques, The 6502 Instruction Set (which includes a complete description of each instruction, in alphabetical order, as a good reference manual would), Addressing Techniques, Input-Output Techniques, Input-Output Devices, Application Examples, Data Structures, Program Development and Conclusion. In addition, there are six short appendices giving some tables (ASCII code, hexadecimal conversion, etc.) which are traditionally found in a book of this nature.

Whereas Foster's book is problem oriented, Zaks' book is solution oriented. Each of the above-named chapters describes some aspect of programming or some feature of the 6502 and then applies this new knowledge to several simple problems. Most chapters also contain additional exercises for the reader, and answers for some of the exercises are given at the end of the chapter. (Answers to more of the exercises would have improved the book, especially for those who are trying to learn on their own and who don't have a teacher to whom they can readily turn in case of difficulty.)

Zaks differentiates well between "programming" and "coding." Programming is the entire process of devising a solution to a problem (i.e., finding an algorithm) and converting that algorithm into a sequence of instructions in some computer language, while coding refers only to the actual writing of those instructions. In other words, programming *includes* coding (and flowcharting, documenting, etc.). Unfortunately, most novice "programmers" see coding as the epitome of programming and, as a result, never become good programmers.

On the negative side, Zaks' book contains a lot of typographical errors, some of which are not listed on the errata sheet provided by the publisher. Although most of them are rather trivial (substitution of the word "accumulator" for "stack pointer" in the explanation of the TXS instruction on page 170, for example), they can be confusing to the beginner.

Both of the 6502-programming books have their good and bad points. If you know almost nothing about the 6502, or if you've never programmed anything before, then Zaks' book is probably the better one for you. On the other hand, if you are an "old-timer" programmer, or if you have already studied the 6502 manuals and written a few short programs, then Foster's book may give you the additional experience you desire.

Myron Calhoun
Manhattan KS

The User's Guide to North Star BASIC
Robert R. Rogers
Interactive Computers
Houston TX, 1978, 241 pages
$14.95

North Star users have long suffered with the "explanation" of North Star BASIC, as provided by North Star Computers, Inc. While that documentation does not pretend to be a complete instruction manual, users might have hoped that North Star could offer a *few* suggestions as to how that BASIC dialect could be used. Although I have some background in BASIC programming, there are many facets of BASIC in general, and my North Star dialect in particular, that I have not understood, or at least have not been able to utilize effectively or efficiently.

Comes, now, Robert R. Rogers, with *The User's Guide to North Star BASIC.* Rogers has assembled a useful manual that explains a great deal about North Star BASIC, specifically, North Star BASIC, Release 3, as applied to Processor Technology's Sol hardware. Variations involving Release 4 are specifically treated at the end of the book. Thirty-six chapters (albeit several only one or two pages long) take a person from rank beginner to a competent intermediate North Star BASIC user.

The peculiarities of North Star BASIC, including the way the disk operating system (DOS) interacts and affects the way the user carries out his activities, are covered thoroughly, almost to the point of distraction.

Rogers begins on an encouraging note by providing a spiral-bound book that lies open, flat on the table. Looking inside, one finds a first chapter dealing with "How do we get *started*?" When Rogers tells me that "You are the master of the plug," and that "All computers, like all cars, have the same knobs," I am ready to listen. He identifies the steps necessary to turn on the system, and discusses what should happen at each step—for example, "Press the square red button . . . you should hear what sounds like a slight thump and then a hummmmm. That's the cooling fan." For those of us who are

For your SWTP 6800 Computer...

PERCOM's™ FLOPPY DISK SYSTEM

the

LFD-400

Ready to plug in and run the moment you receive it. Nothing else to buy, no extra memory. No "booting" with PerCom MINIDOS-PLUSX™, the remarkable disk operating system on EPROM. Expandable to either two or three drives. Outstanding operating, utility and application programs.

To order or request literature call Percom toll-free 1-800-527-1592.

only **$599.95**

fully assembled and tested shipping paid

PERCOM™

PERCOM DATA COMPANY, INC.
211 N. Kirby Garland, TX 75042
(214) 272-3421

For the low $599.95 price, you not only get the disk drive, drive power supply, SS-50 bus controller/interface card, and MINIDOS-PLUSX™, you also receive:
• an attractive metal enclosure • a fully assembled and tested interconnecting cable • a 70-page instruction manual that includes operating instructions, schematics, service procedures and a complete listing of MINIDOS™ • technical memo updates — helpful hints which supplement the manual instructions • a 90-day limited warranty.

SOFTWARE FOR THE LFD-400 SYSTEM

Disk operating and file management systems

INDEX™ The most advanced disk operating and file management system available for the 6800. INterrupt Driven EXecutive operating system features file-and-device-independent, queue-buffered character stream I/O. Linked-file disk architecture, with automatic file creation and allocation for ASCII and binary files, supports sequential and semi-random access disk files. Multi-level file name directory includes name, extension, version, protection and date. Requires 8K RAM at $A000. Diskette includes numerous utilities $99.95

BASIC Interpreters and Compilers

SUPER BASIC A 10K extended disk BASIC interpreter for the 6800. Faster than SWTP BASIC. Handles data files. Programs may be prepared using a text editor described below $49.95
BASIC BANDAID™ Turn SWTP 8K BASIC into a random access data file disk BASIC. Includes many speed improvements, and program disk CHAINing $17.95
STRUBAL+™ A STRUctured BAsic Language compiler for the professional programmer. 14-digit floating point, strings, scientific functions, 2-dimensional arrays. Requires 20K RAM and Linkage Editor (see below). Use of the following text editors to prepare programs. Complete with RUN-TIME and FLOATING POINT packages $249.95

Text Editors and Processors

EDIT68 Hemenway Associates' powerful disk-based text editor. May be used to create programs and data files. Supports MACROS which perform complex, repetitive editing functions. Permits text files larger than available RAM to be created and edited $39.95

TOUCHUP™ Modifies TSC's Text Editor and Text Processor for PerCom disk operation. ROLL function permits text files larger than available RAM to be created and edited. Supplied on diskette complete with source listing $17.95

Assemblers

PerCom 6800 SYMBOLIC ASSEMBLER Specify assembly options at time of assembly with this symbolic assembler. Source listing on diskette $29.95
MACRO-RELOCATING ASSEMBLER Hemenway Associates' assembler for the programming professional. Generates relocatable linking object code. Supports MACROS. Permits conditional assembly $79.95
LINKAGE EDITOR – for STRUBAL+™ and the MACRO-Relocating assembler $49.95
CROSS REFERENCE Utility program that produces a cross-reference listing of an input source listing file $29.95

Business Applications

GENERAL LEDGER SYSTEM Accommodates up to 250 accounts. Financial information immediately available — no sorting required. Audit trail information permits tracking from GL record data back to source document. User defines account numbers $199.95
FULL FUNCTION MAILING LIST 700 addresses per diskette. Powerful search, sort, create and update capability $99.95
PERCOM FINDER™ General purpose information retrieval system and data base manager $99.95

™ trademark of PERCOM Data Company, Inc.
™ trademark of Hemenway Associates Company.

Now! The LFD-800 and LFD-1000. Add one, two or three LFD-800 drives and store 200K bytes per drive on-line. Add one or two (dual-drive) LFD-1000 units and store 800K bytes per unit on-line. Complete with interface/controller, DOS, cable & manuals. **Two-drive** systems: LFD-800 — $1549; LFD-1000 — $2495.

✓P7 **PERCOM** ***'peripherals for personal computing'***

convinced that every untried act on our multi-thousand-buck toy will result in executing the undocumented command "Destruct Everything," Rogers provides many such reassuring comments.

A brief discussion of what the DOS is, and the differences between being in DOS, BASIC and "start-up" modes, suggests that each mode has its purposes. Rogers doesn't deal with the start-up mode, other than to tell us how to get out of it.

He then goes on to deal with writing programs and line numbers, listing, how to use variables, the characters available, the various editing features of North Star BASIC, using the DOS to store and retrieve programs and to manipulate the disk contents; then he begins the use of North Star BASIC as a programming language. The peculiar aspects of North Star BASIC, such as the RANDOM and CHAIN commands, numerical formats and others that could create confusion, are discussed at length.

North Star BASIC, of course, is oriented toward North Star's disk system, and has many functions within the language for file activities. Rogers discusses, at a very elementary level, what goes on inside the disk, and how information is applied to the disk. He spends considerable time dealing with data files, their formats and the ways the computer manipulates that data as it goes onto and is pulled off the diskettes. These details have an obvious relation (to any hardware freak) to what the computer can do with the data, so they are essential to effective manipulation of the data. Here, Rogers takes a patient tutoring approach with us neophytes and provides a clear concept of what is actually going on. I found out that I could do a lot of things with my system that North Star didn't tell me about. In fact, over 50 pages deal with sequential and random access to data files.

The last three chapters (34, 35, 36) are remarkable in that they make an attempt to open up some things that nobody will talk about. Chapter 34 deals with hexadecimal vs decimal arithmetic, a subject that is important to proper understanding of memory allocations, manipulations of computer locations, etc. Arithmetic and I do not get along. Now, here is a mathematical system that assumes that I have *eight* fingers on each hand when, after seven handfuls of years, I'm still not completely proficient with *five* per hand! Rogers offers ten pages dealing with converting hex to decimal, and, amazingly, I sort of understand it.

Chapter 35 deals with "secrets" of North Star BASIC. While some of these suggest ways to get "inside" the SOL video display and mess around with making the video screen do things other than print words and numbers, other suggestions deal with changing output from video to other output devices, making the DOS do more than initially offered, etc.

Rogers admits using the "friendly interrogation" technique, which involves copious quantities of foamy-headed "loosener" applied to the local computer-store person, a technique that merits further investigation. (Unfortunately, my nearest computer-store person is 700 miles away.) What he discovered were some of the most fascinating portions of the North Star BASIC capabilities. I only wish that Rogers had been able to read more of his notes the next morning.

The last chapter deals with specific differences between Releases 3 and 4 of North Star BASIC. I understand that there is a Release 5 in the North Star hatchery—I hope that Rogers will make a second effort, updating this guide to reflect Release 5.

What is missing in this book? Not much. I would have liked to see some discussion of the DEF function. Someone once tried to

(continued on page 25)

Release Yourself

A broken tile, a rusty sword, the timeworn remnants of a medieval tapestry: Nothing sparks the imagination more than these fragile reminders of another place, another time, another way of life . . . nothing, that is, save the prospect of living it. That's why Santa Paravia and Fiumaccio is fast becoming a national fad among avid computer gamers and computer professionals alike. Why not get in on the fun today?

This superb simulation casts up to six lords and ladies as competitors in a race for the rule of their respective 15th century Italian city-states.

Each player begins with 1,000 Florins, a small parcel of land, a suitable complement of serfs, and a prewritten obituary and time of death. Life was short back then, so you'd better move quickly . . . from this point on you're on your own.

HARVEST

The winter ended, the grain steward reports the minimum grain amount necessary to feed your people; depending on how much of your reserves were eaten by rats, you may need to purchase additional grain. Underfeeding will cause many serfs to die, and will also severely lower the birth rate. If you fared the winter well, you may be in a position to sell surplus grain or to overfeed your people, which will cause a higher birth rate and attract serfs from less fortunate neighboring towns—something you should consider if you plan to purchase additional land for farming.

You can also speculate in land and grain at this time. The price may vary from year to year, but an average increase is certain.

TAXES

You must now decide the levels of taxation for your community. As in real life, the consequences of your decisions are far-reaching: Set the customs tax too high, and businesses will suffer; a high income tax won't sit well with the wealthy; and an overburdening sales tax has got to cut down receipts at your markets. Experiment, and you're sure to find a policy that will work well with your present economy—but changes will surely be needed as your community grows.

You've also got to decide a policy for justice. Will you be very fair (costs money, but is great for the economy) or outrageous (taking bribes, selling justice to the highest bidder)? A lenient stance will attract more serfs, while outrageous justice will soon have them fleeing to more pleasant surroundings.

MAP PHASE

The computer will now draw a detailed map of your area. From this map you will be able to determine the adequacy of your defenses, the ratio of workers to acreage, the number of woolen mills and market places, and the size of your castle and cathedral.

PUBLIC WORKS

Your treasury laden with the fruits of a year's labor, you can now purchase a woolen mill or two, or invest in more market places. Maybe you'll decide to increase the size of your castle. If you feel that more clergy support might hasten your rise to the throne, build another wing on your cathedral. If your land area has grown quickly, this might be the time to arm another unit of serfs for your regions' defense. Your computer will now look back at what you've accomplished in the last year, and decide if you merit a higher title.

OBITUARY

At this point, the computer will check to see if you've reached "the fullness of time." If so, it will print the year and cause of your death, and your highest rank obtained. Although the computer will no longer offer your turn, the statistics of your reign will be kept in the comparison table until the game's end. Since it's whoever reaches the throne first, or achieves the highest title before death who wins . . . you could still wind up the winner. In any case, you're sure to end the game a little wiser—and chomping at the bit to play again.

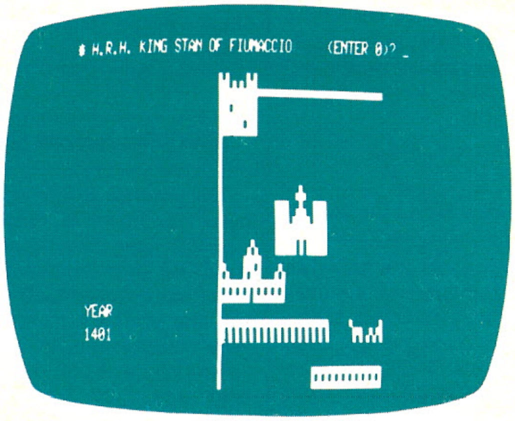
Available for the 16K level I and level II TRS-80 microcomputer. Order No. 0043R. $7.95.(See pg. 107 for order blank)

INSTANT SOFTWARE™ $7.95

Designed for use on TRS-80* 16K LEVEL I LEVEL II

Santa Paravia and Fiumaccio

* A trademark of Tandy Corporation



0043R Instant Software Inc., Peterborough NH 03458 USA *See reverse for program information*

Instant Software™ Inc.

Peterborough, N.H. 03458—603-924-7296

NEW PRODUCTS

Edited by Dennis Brisson

AIM16

Now you can let your computer listen to the real world with the AIM16, the analog input module of the DAM (Data Acquisition Modules) Systems from Connecticut Microcomputer, 150 Pocono Rd., Brookfield CT 06804. The AIM16 has sixteen 8-bit analog inputs. Each input is individually addressed. Conversion time is 100 microseconds.

The AIM16 can be used with any computer that has an 8-bit output port and an 8-bit input port. It sells for $159. The AIM16 starter kit (AIM16, power supply and input and output connectors) sells for $189. Reader Service number C107.

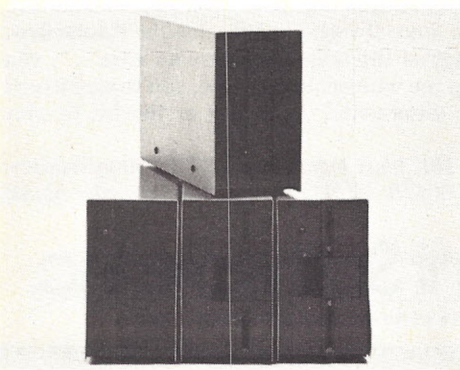
Add-on Disk Drives for TRS-80

TRS-80 owners now have their choice of single-head or dual-head add-on disk drives. Microcomputer Technology, Inc. (MTI), 2080 S. Grand Ave., Santa Ana CA 92705, is manufacturing the first family of add-on disk drives for the TRS-80.

The MTI single-head disk drive family (TF-X) offers a choice of MPI, Pertec or Shugart SA400 mini-floppy disk drives. Shugart is the same device offered by Radio Shack, while the Pertec provides quieter operation and the use of the Flippy diskette (uses both sides). The MPI unit provides additional features, such as door lock and automatic diskette ejection, normally found in the larger 8-inch disk drives. Prices for the TF-X single-head units start at $379.

MTI's dual-headed units (TDH-X) provide the same capacity as two single-headed drives at a substantial savings in space and money. The TDH-X units are priced at $675.

Interfacing the MTI add-on disk drives to the TRS-80 is accomplished via the TRS-80 expansion interface, which can accommodate up to four single-headed drives or two double-headed drives. Operating software is available from Radio Shack. Reader Service number M82.

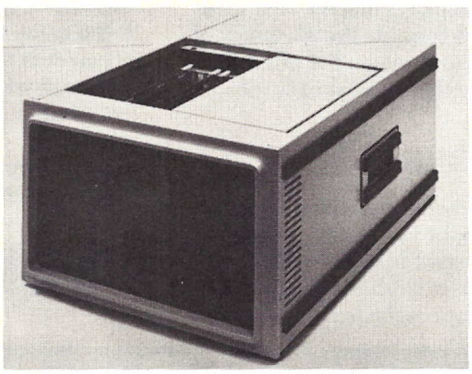


MTI disk drives.

AmSYS Development System

The AmSYS 8/8 Microcomputer Development System is especially designed to support the AmZ8000 16-bit microprocessor in both hardware and software. It will also provide complete software support and an upward-compatible path for the 8080, 8085 and Z-80 8-bit microprocessors.

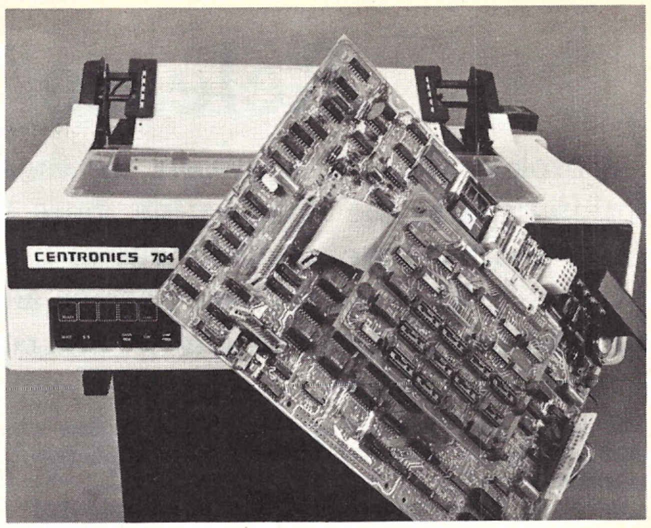
Advanced Micro Computers, 3340 Scott Boulevard, Santa Clara CA 95051. Reader Service number A91.



The AmSYS 8/8.

1200 Baud Communications Printer

The Model 704 data communications printer combines high throughput, flexible communications capabilities and outstanding print features to provide an unconventional set of end-user advantages. High throughput from 70 to 400 lpm in the Model 704 is achieved via the printer's 180 cps print speed, which is capable of infinitely sustaining a 28-character line length at 1200 baud. Of the device's flexible communications features, the Model 704's built-in RS-232C serial interface and operator-selectable line protocols are especially noteworthy. And, excellent print quality is provided in the Model 704 through such standard features as a nine-pin free-flight print head and a full 96-character ASCII set.



The Model 704.

Three line protocols—XON/XOFF, reverse channel and Data Terminal Ready—provide an unusual degree of printer interaction with host operations. Switch selectability is enhanced by the convenient location of the 704's switch panel just behind the printer's front panel. A nine-foot RS-232C cable comes standard with the printer, while a 2K page buffer can be added as an option to increase system efficiency. The printer carries an end-user price as low as $2350.

Centronics Data Computer Corp., Hudson NH 03051. Reader Service number C136.

M6800 Microsoft BASIC

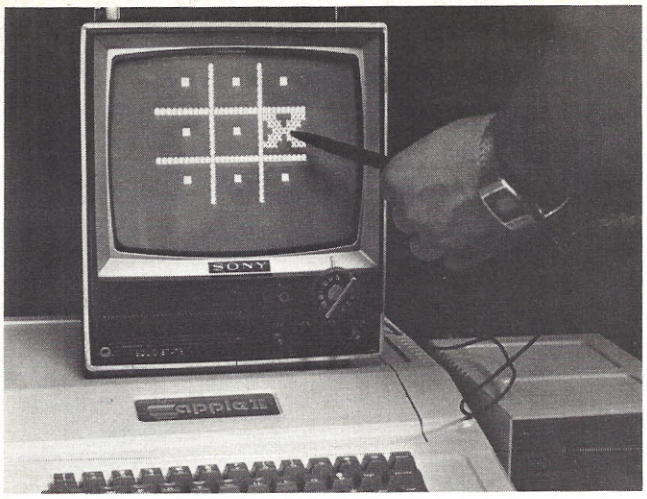
The new M6800 version of Microsoft BASIC allows users of M6800-based microcomputers to take advantage of the large existing library of Microsoft BASIC programs. All of the features of the 8080 BASIC have been implemented, including error trapping, edit mode, random access files, renumbering, 16-digit accuracy, full PRINT USING and IF/THEN/ELSE.

The commercial use of M6800-based computers will be greatly enhanced by the flexibility and features of Microsoft BASIC. Business applications software written in Microsoft BASIC is already available from several sources. Requiring less memory than any other BASIC, M6800 meets the ANSI standard for BASIC and is available in three upward-compatible versions: 8K BASIC, Extended BASIC and Disk BASIC.

Microsoft, 10800 NE Eighth, Suite 819, Bellevue WA 98004. Reader Service number M78.

Apple II Light Pen

A low-cost light pen is now available for simple installation and immediate operation in Apple II applications such as bar graphs, charts and games. The Apple II Light Pen is supplied with three demonstration programs on cassette. These demo programs not only exemplify the many uses of the pen, but may also aid in developing BASIC



Light pen at work.

programs to drive the pen.

The first demonstration program instructs on the use of the light pen as a menu-selection tool. Second, a program of graphics demonstrations permits the user to select from a menu of graphic shapes and colors. Selection from either the shape or color menu is accomplished by depressing the RETURN key. The third program is a graphics color bit-pad demonstration. A color may be selected from the color menu by depressing any key.

The pointer software driver performs seven functions, which include selection of graphics mode and page two display, search for x and y ordinates, a test for odd/even y ordinate, set page one display and return to calling program. The entire package, including light pen, software on cassette and operating manual, costs $34.95.

Programma International, Inc., 3400 Wilshire Blvd., Los Angeles CA 90010. Reader Service number P48.

Z8000 CPU Circuit

The Z8000 microcomputer CPU circuit offers users in a single-chip device the architectural resources of mini- and large, mainframe computers. The Z8000 CPU is available in two versions—the Z8001, in a 48-pin ceramic DIP that allows the user to address up to 8 megabytes of memory, and the Z8002, in a 40-pin ceramic DIP.

The 40-pin Z8002, designed for smaller, less memory-intensive applications, is compatible with the 48-pin Z8001, but the 40-pin CPU's addressing is limited to 64K bytes in each of its 6 address spaces. A scaled N/MOS depletion-load silicon-gate device, the Z8000 CPU densely packs 17,500 transistors on a chip 238 by 256 mils. The Z8001 costs $195; the Z8002 sells for $150.

The Z8000, designed for both minicomputer and microcomputer applications, features 24 on-chip 16-bit registers that reduce the number of memory references needed in programming. Sixteen of those registers are general-purpose. The Z8000's problem-solving instruction set supports seven different data types from single bits to 32-bit words and has eight addressing modes and 418 usable opcode combinations.

Zilog, Inc., 10340 Bubb Rd., Cupertino CA 95014. Reader Service number Z3.

Expanded Dual-Floppy Storage

The CD2+2 option provides a big step up from the storage limitations of the standard double-floppy system at a great cost advantage over an alternative hard disk drive. With the CD2+2 option, Ohio Scientific C3 systems operating under the OS-65U disk operating system will gain an increase in storage availability from the approximately 275K bytes per disk surface of the standard double-floppy system to 1.1 megabytes of storage. By utilizing both sides of the magnetic medium, the CD2+2 option doubles storage capacity. OS-65D DOS users will experience similar storage increases to about 1 megabyte.

CD2+2's increased storage also results in time savings. The ability, by virtue of increased storage capacity, to store multiple copies of working files on various operating disks can reduce or eliminate the numerous disk-swapping operations that might otherwise be required. This advantage increases the productivity in business inventory and data-base management operations. The CD2+2 option is offered for use with Ohio Scientific's models C3-OEM and C3-A. The CD2+2 costs $1200.

Ohio Scientific, 1333 Chillicothe Rd., Aurora OH 44202. Reader Service number O1.

PET Hard-Copy Interface

Now you can produce hard copy with your PET computer with the hard-copy interface unit from GPA Electronics, Inc., PO Box 7410, Oakland CA 94601. The PET interface device uses the IEEE-488 bus. Parallel signals are converted to serial signals by a UART. All logic signals are converted to the proper levels. For output, there is a standard DB-25 female connector.

But best of all, you still retain the IEEE-488 bus. GPA provides you with an IEEE edge connector that allows you to extend the bus for additional devices. The unit will selectively recognize primary address #5. The interface device has selectable baud rates (110, 300, 600, 1200, 2400, 4800, 9600), as well as both RS-232C and 20 mA, and current loop outputs. Reader Service number G24.

Dibex Timesharing System

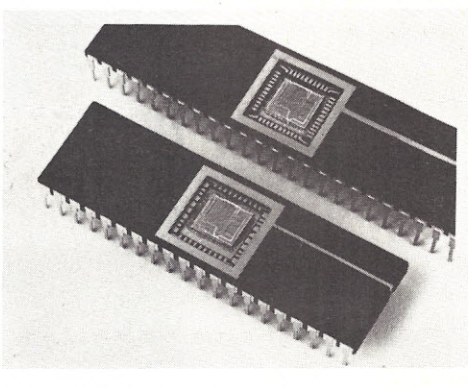
You can now use your H11 Heathkit computer as a small-business computer with the Dibex VM Timesharing Operating System. The system has a DIBOL language compiler that enables access to any of the existing application packages written in this previously unavailable DEC language. Through the DIBEX Virtual Memory techniques, you can get up to 30 percent or more jobs running concurrently. Additional functions can be added to the system, as required, without a corresponding increase in hardware.

If you want to do your own programming, this language is easy to use, especially since you'll be able to develop while someone else runs applications. The Multi-Terminal Development allows the concurrent use of Editing, Compiling and Linking, along with other utilities. Through the addition of an IF/ELSE, COPY/INCLUDE and a WHILE STATEMENT, the expanded language elements make programming business applications easier. Index Sequential Access Method (ISAM), SORT, Line Printer Spooler and a Message utility are available as options. The base system requires at least 56K, an I/O device, floppy disk and $1995.

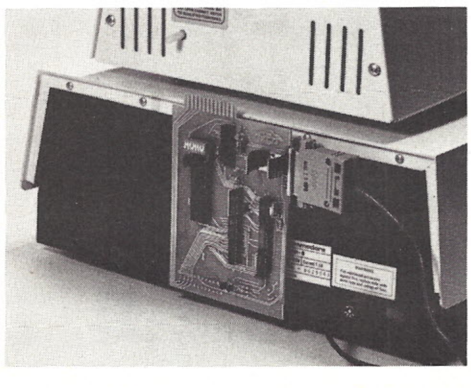
Information Access Systems, Inc., 1129 Bloomfield Ave., West Caldwell NJ 07006. Reader Service number I38.

Kable Kraft Cables

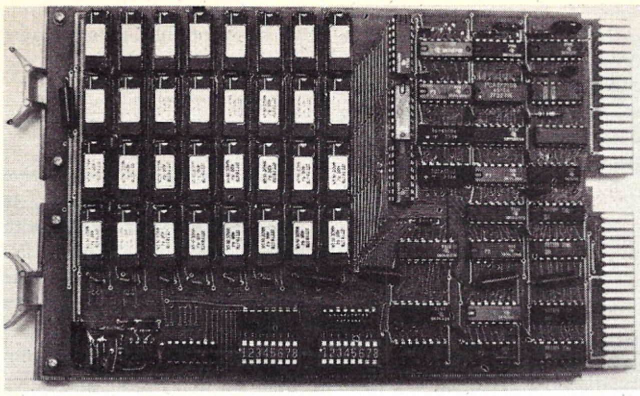
The Kable Kraft line of high-quality, preassembled data communications cables configured



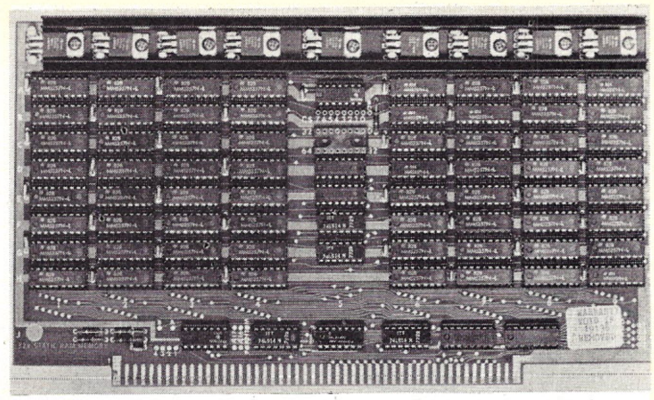
The Z8001 (top) and the Z8002 (bottom).



GPA interface for the PET.



The CI-1103.



Tarbell 32K board.

for EIA standard RS-232C applications are available in standard nine or eighteen foot lengths and are nine conductor and color coded. Heat-shrink tubing surrounds each individual connection for added strength and protection in order to preserve data integrity. All connectors are standard 25 pin. Kable Kraft cables are also available in custom lengths and configurations upon special request.

CompuCable Corporation, 2081 Business Center Drive, Suite 180, Irvine CA 92715. Reader Service number C137.

Inflation Beaters

National Software Marketing, Inc., 4701 McKinley St., Hollywood FL 33021, announces the release of Inflation Beaters, designed to run on the TRS-80. Statistics Pack-1 has two modules. The first, Linear Regression, reads a distribution of paired x-y values provided by the user. The output contains the paired values mean of x and y, the standard deviation of x and y and an expected value of y for any given x. The other module, Correlation, reads in pair x-y values and outputs paired values, correlation coefficient, observation count, mean variance and standard deviation of x and y.

Statistics Pack-2 reads a distribution of paired x-y values, calculates expected values and outputs the chi-square value and degree of freedom. Both observed and expected value matrices are displayed. Statistics Pack-3 produces for each distribution value its z score equivalent sorted in ascending order. Also printed is the sum of squares, variance, median, standard deviation and skewness.

Statistics Pack-4 calculates chi-square, slope for linear regression, mean, variance and standard deviation and T-ratio for two groups of unpaired data. Statistics Pack-5 computes multiple linear regression, variance tables for analysis of Greco-Latin square and calculates F ratio for Youden square design. Statistics Packs 1, 2 and 3 cost $12.95; Statistics Packs 4 and 5 are $12. Reader Service number N21.

64K Memory for the H11

The CI-1103 is designed specifically for the Heathkit/Digital H11 computer and LSI 11/2 and PDP 11/03 microcomputers. The new memory circuit features an 8K × 16 dual-width board using 200 ns 4027 4K × 1 dynamic memory devices or a 32K × 16 dual-width board using 200 ns 4116 16K dynamic memory devices. The unit simply plugs directly into the Heathkit/Digital H11, LSI-11, PDP 11/03 or the LSI-11/2.

The CI-1103 is available with either on-board distributed refresh or external refresh control logic. Data access time is 300 ns, and cycle time is 525 ns. On-board memory select is available in 2K increments up to 128K words of memory. The CI-1103, 8.44 × 5.187 inches, comes available with battery backup capability. Power consumption is under 7 Watts. Price for the 8K × 16 is $390; the 32K × 16 is $750.

Chrislin Industries, Inc., 31312 Via Colinas #102, Westlake Village CA 91361. Reader Service number C125.

32K RAM Board

The new Tarbell 32K static RAM board is S-100 bus compatible and runs at 300 ns. It features extended addressing, or bank switching, and contains nine regulators that greatly enhance its heat distribution. A phantom line for disabling memory and for such functions as bootstrapping is included. Because only high-quality ICs are used, power requirements are minimal. The same board is available with 16K, leaving half of the board open for future addition of chips.

The Tarbell memory board is supplied with a 20-page operating manual that includes a complete parts inventory, schematics and several test routines. The 32K RAM board, assembled and tested, is priced at $625; the 16K version, assembled and tested, is $390.

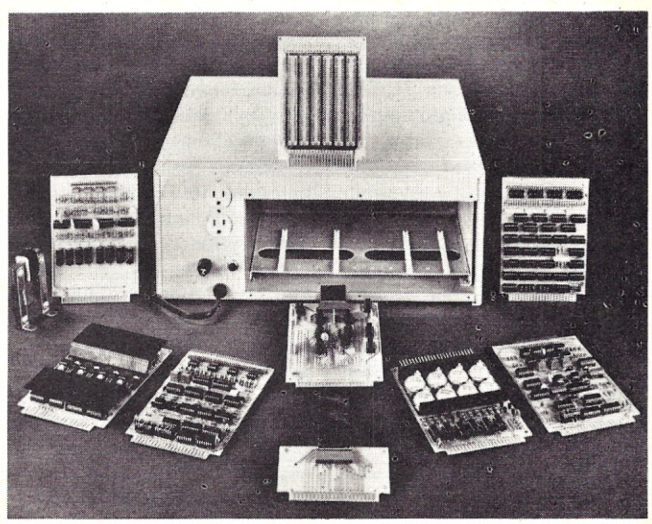
Tarbell Electronics, 950 Dovlen Place, Suite B, Carson CA 90746. Reader Service number T11.

Data Acquisition and Process Control System

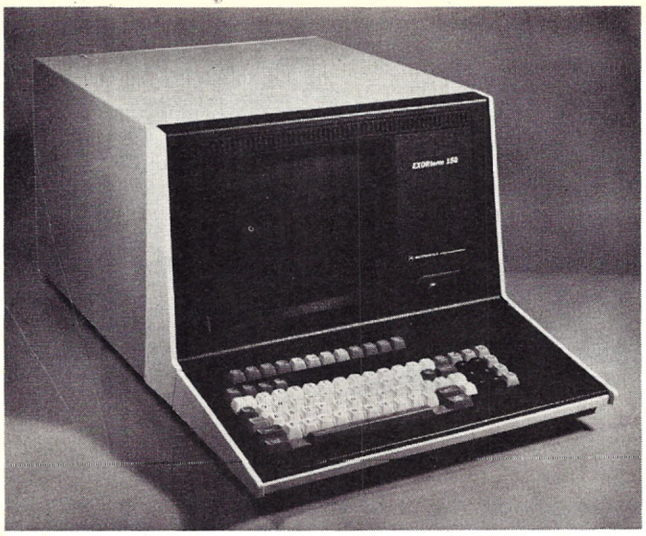
The Real World Interface System is the first low-cost, general-purpose data acquisition and process control system designed for use with mini and microcomputers. It features many different plug-in modules that give the user the flexibility and power to configure his system according to his needs. Applications include environmental control, peak demand limiting, robotics and automated assembly-line testing.

The Real World Interface System has its own cabinet, which includes a power supply, card cage and motherboard with slots for up to 12 plug-ins. The plug-in modules include A/D and D/A converters and a computer interface card. Each plug-in card has at least eight channels of input or output (except for a maximum of four channels on the current probe); some have 16 or even 32 channels.

Most plug-ins are off the shelf, and all plug-ins can be purchased in either kit form or assembled and tested. The modules range in price from $65 (kit) ($79.50, assembled) for the 8-channel dc controller card to $125 (kit) ($150, assembled) for the 8-channel ac controller card. A



Real World Interface System.



EXORterm 150.

cabinet, complete with motherboard, power supply and parallel CPU interface, is $299 (kit) ($360, assembled).

General Computer Technology, 400 South Lipan, Suite 2, Denver CO 80223. Reader Service number G30.

Motorola Display Terminal

EXORterm 150 is an M6800-based display system using predominately LSI components of the M6800 family. It provides control of the display attributes, communication facility, terminal switch/indicator control and keyboard inputs. Micro-executive firmware, in conjunction with control and application task firmware, coordinates the functions of EXORterm 150 in its EXORciser-oriented activity.

The EXORterm 150 has been upgraded from the EXORterm 100 in that it has an improved keyboard that adds the cursor control, tabbing, page, line and character control keys. As the terminal and console to be used with the EXORciser I or II, the EXORterm 150 now provides for a total Motorola system, including the improved keyboard, which allows use of Motorola's new CRT Editor.

As an EXORciser display console, EXORterm 150 facilitates the exchange of data between the user and the system via a high-quality video interface, keyboard entry and a serial communications link using speeds up to 9600 bits per second. It consists of the following elements:

- Motorola M3000 video monitor for display
- CRT controller board containing all of the necessary control electronics and firmware operating routines
- CRT configuration board providing the means to manually select basic operating criteria
- Chassis/housing with power supply
- Keyboard for data entry
- Necessary resident executive firmware to control the display and communications interface.

Price of the basic EXORterm 150, EXORciser display console (110 V, 60 Hz option) is $2490. The keyboard only is $340; the ROM set is $200; and the new CRT Editor (M6800EDITORM) is $300.

Motorola Microsystems, PO Box 20912, Phoenix AZ 85036. Reader Service number M20.

64K RAM Board

The ZS-Systems 64K RAM board is designed to operate in any Z-80-based S-100 bus computer. It uses 16K dynamic RAM chips and features board select, bank select, transparent on-board refresh, 2 MHz or 4 MHz operation (with no wait state) and memory disable. It is compatible with the Cromemco system. Fully assembled, burned in and tested, the board sells for as low as $500.

The board is used in the ZS-800 microcomputer from ZS-Systems, PO Box 1847, San Diego CA 92112. The ZS-800 is available in two versions: the ZS-800 and the ZS-800E, which includes the Z-80A microprocessor, 64K memory expandable up to one-half megabyte, four expandable I/O ports and a modem to assure easy accessibility through the existing communication network.

ZS-800 computers feature compatibility with most major computer systems using IBM format on Shugart floppy disk drives, low-power consumption and expansion using the state-of-the-art 16K memory ICs and a CP/M disk operating system. Reader Service number Z4.

FSK-80

TRS-80 owners, now you can solve all your cassette problems, including problems of volume level, with the FSK-80. The FSK-80 takes binary information coming from the TRS-80 with an impulsion and converts it to a frequency modulated signal (frequency shift keying mode). Because the only information used is the frequency, it is possible to read recorded data with a good signal/noise. The reading succeeds with any volume level (from 1 to 10).

The FSK-80 has two ports: the cassette adapter and an acoustic modem extension. The cassette adapter has a switch that allows you to choose the normal or FSK mode. You can make the normal-to-FSK tape conversion with the following operations:

CLOAD of a program with the switch on normal
CHANGE cassette
CSAVE with switch on FSK
It is also possible to transform binary programs.

With the acoustic modem extension you can transmit data and programs on a phone line with the alternate mode. The modem extension, which also has a LOCAL/LINE switch, is connected to the cassette adapter with a 5 DIN cable. Approximate costs for the cassette adapter and modem extension are $150 and $70, respectively.

Micrologiciels, Sarl au capital de 2000 FF 160, rue Etienne Dolet, 94140, Alfortville France. Reader Service number M91.

TRS-80 Case

Now you can provide complete protection for your TRS-80 with the C1 case from Designco, PO Box 307, Union MI 49130. The case comes in kit form and, when assembled, encases the separate modules and interconnecting cables securely and efficiently for a permanent work location or allows safe portability if movement of the computer system is required.

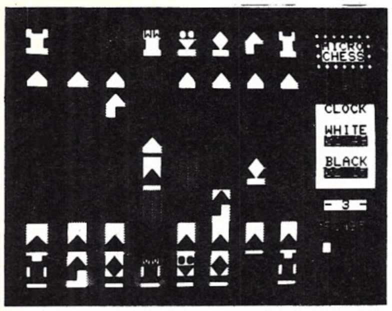
The 14 × 19 × 25 inch case is made of black ABS thermoformed side panels with exposed wood sections covered by a silver, quilted, leather-like material, giving the completed unit a custom-made, whole-system appearance. Assembling the unit with just a Phillips screwdriver is facilitated with predrilled holes, uniform screw size and comprehensive instruction. The unit weighs 9 lbs. and costs $69.50, plus $5 shipping. Reader Service number D47.

Apple II Disk Text Editor

EDIT, a DOS text editor for the Apple II microcomputer, was designed to facilitate changes to disk files, but input and output via cassette is also supported. The text editor includes 25 commands and will edit fixed or variable-length disk files. System commands allow the user to Delete, Insert, Change, Display, Add and Print records. String commands facilitate searching and changing part of a record or the entire file. User-defined tabs, file

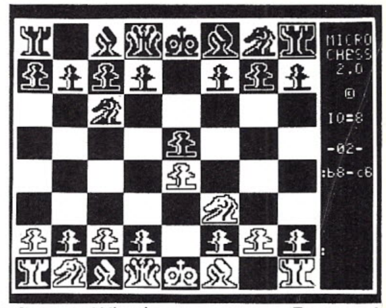
The C1 case for the TRS-80.

PET / TRS-80 / APPLE: Personal Software brings you the finest!

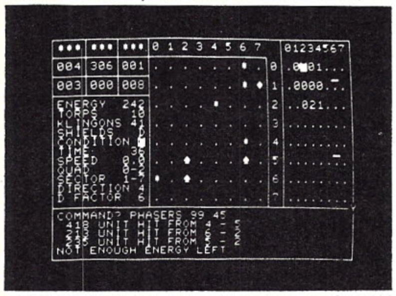


MICRO CHESS

The Industry's First Gold Cassette Over 50,000 Sold

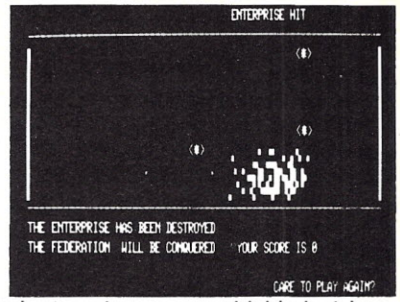


MICROCHESS is the industry's best selling computer game. And no wonder—because MICROCHESS gives you more than just a chessplaying program: A convenient, foolproof set of commands and error checks ... complete instructions in a 5½" by 8½" booklet ... a cassette that's guaranteed to load, with disk versions coming soon ... and several levels of difficulty to challenge you not just once, but time after time. It's available through well over three hundred computer stores and many mail order sources ... always originating from Personal Software. What's more, *every* Personal Software product is selected to give you these same benefits of easy availability, reliable cassettes, readable documentation, a carefully thought out user interface ... and most important, continuing challenge and enjoyment, not just once but time after time. If you haven't already, order your own gold cassette: MICROCHESS, by **Peter Jennings**, for 8K **PET**s, 16K **APPLE**s, and 4K Level I and II **TRS-80s** **$19.95**

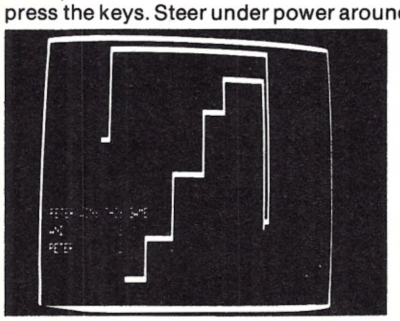


TIME TREK

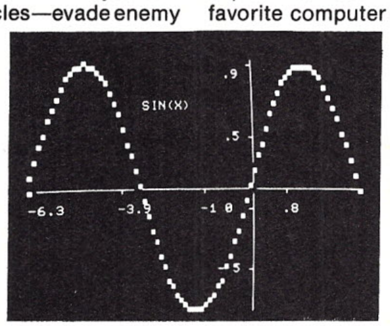
A Tour De Force In Real Time Action Strategy Games



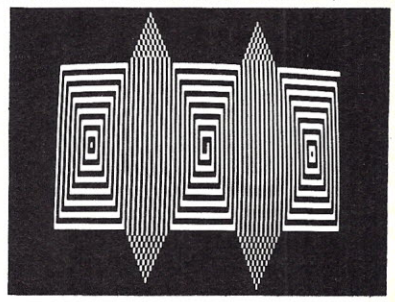
TIME TREK by **Brad Templeton** for 8K **PET**s and **Joshua Lavinsky** for 4K Level I and II **TRS-80s** adds a dramatic new dimension to the classic Star Trek type strategy game: REAL TIME ACTION! You'll need fast reflexes as well as sharp wits to win in this constantly changing game. Be prepared—the Klingons will fire at you as you move, and will move themselves at the same time, even from quadrant to quadrant—but with practice you can change course and speed, aim and fire in one smooth motion, as fast as you can press the keys. Steer under power around obstacles—evade enemy shots as they come towards you—lower your shields just long enough to fire your phasers, betting that you can get them back up in time! With nine levels of difficulty, this challenging game is easy to learn, yet takes most users months of play to master. ADD SOUND EFFECTS with a simple two-wire hookup to any audio amplifier; the TRS-80 also produces sound effects directly through the keyboard case, to accompany spectacular graphics explosions! You won't want to miss this memorable version of a favorite computer game **$14.95**



BLOCKADE by **Ken Anderson** for 4K Level I and II **TRS-80s** is a real time action game for two players, with high speed graphics in machine language. Each player uses four keys to control the direction of a moving wall. Try to force your opponent into a collision without running into a wall yourself! A strategy game at lower speeds, BLOCKADE turns into a tense game of reflexes and coordination at faster rates. Play on a flat or spherical course at any of ten different speeds. You can hear SOUND EFFECTS through a nearby AM radio—expect some razzing if you lose! **14.95**



GRAPHICS PACKAGE by **Dan Fylstra** for 8K **PET**s includes programs for the most common 'practical' graphics applications: PLOTTER graphs both functions and data to a resolution of 80 by 50 points, with automatic scaling and labeling of the axes; BARPLOT produces horizontal and vertical, segmented and labeled bar graphs; LETTER displays messages in large block letters, using any alphanumeric or special character on the PET keyboard; and DOODLER can be used to create arbitrary screen patterns and save them on cassette or in a BASIC program **$14.95**



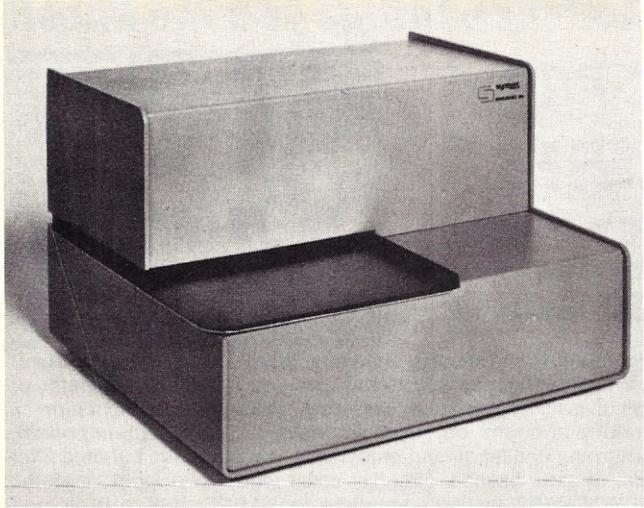
ELECTRIC PAINTBRUSH by **Ken Anderson** for 4K Level I and II **TRS-80s**: Create dazzling real time graphics displays at speeds far beyond BASIC, by writing 'programs' consisting of simple graphics commands for a machine language interpreter. Commands let you draw lines, turn corners, change white to black, repeat previous steps, or call other programs. The ELECTRIC PAINTBRUSH manual shows you how to create a variety of fascinating artistic patterns including the one pictured. Show your friends some special effects they've never seen on a TV screen! **$14.95**

WHERE TO GET IT: *Look for the **Personal Software™** display rack at your local computer store. If you can't find the product you want, you can order direct with your VISA/Master Charge card by dialing **1-800-325-6400** toll free (24 hours, 7 days; in Missouri, dial 1-800-3426600). If you have questions, please call **617-783-0694**. Or you can mail your order to one of the addresses below, as of the dates shown.*

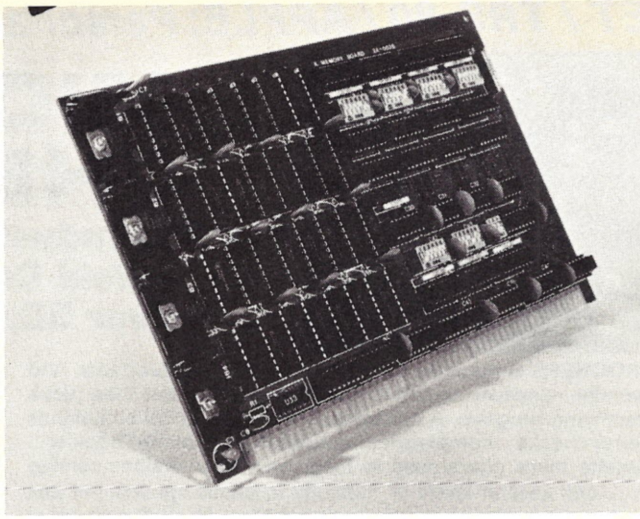
Until July 1: P.O. Box 136
Cambridge, Mass. 02138

P39 **Personal Software™**

After July 1: 592 Weddell Dr.
Sunnyvale, Calif. 94086



The SP-308.



Gimix 16K RAM board.

concatenation, range and other commands are also included.

EDIT is written in Applesoft II Extended BASIC and requires 16K of memory with an Applesoft ROM or cassette-only version; otherwise, a minimum of 24K is suggested. EDIT is provided on cassette or Apple II diskette, complete with user manual. Price is $16.95 (add $5 if on diskette and state if Applesoft ROM).

Service Unique, Inc., 2441 Rolling View Dr., Dayton OH 45431. Reader Service number S93.

Ticket Printer

The SP-308 Ticket Printer is a stand-alone unit that mates the power of the microprocessor and a reliable print mechanism to provide an economical method of ticket printing. The 40-column 5 × 7 dot matrix, impact printer accepts multi-copy forms up to 11 × 17 inches. Printing is at the rate of 50 cps. The microprocessor controller accepts ASCII data in either RS-232-C or 20 mA current loop formats. Standard data rates to 9600 baud are available. Parity and number of stop bits are switch programmable.

The SP-308 provides 40-character buffering (120 characters optional), double width print capability, tab functions and pressure roll release control. Additional features include jumper-selectable print intensity, standard 25-pin EIA input connector and printer busy signal. The SP-308 printer is 11.75 × 7.25 × 9.25 inches. Power requirements are 110/220 V ac 50/60 Hz @25 Watts. Price is $678.

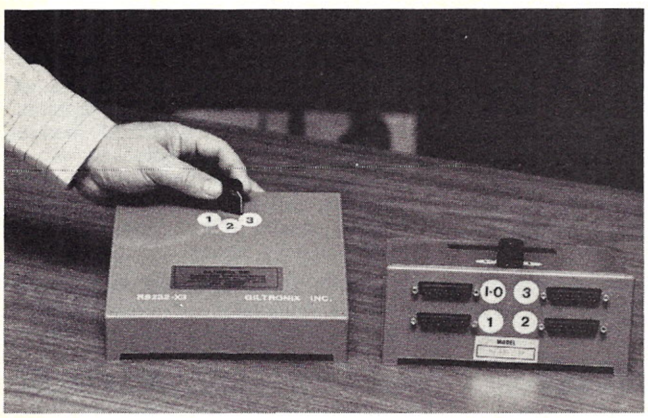
Syntest, 169 Millham St., Marlboro MA 01752. Reader Service number S94.

RS-232 Peripheral User

The RS-232-X line of miniature switching units from Giltronix, Inc., 3156 Avalon St., Palo Alto CA 94306, allows you to connect peripherals, modems and CRTs in any conceivable configuration. After the configuration is established, by a turn of a switch, various peripherals can be selected or deselected.

The most popular unit, the RS-232-X3, can selectively connect your printer, CRT, modem, etc., to the main driving device (mini, micro or main CPU). Another common application for the RS-232-X3 is to select from two or more driving devices the device that will be connected to a printer, CRT, etc. In this case, the printer (or peripheral) will be shared by the driving devices and, therefore, reduce the overall system cost.

The unit price for the RS-232-X3K (kit version) is $47.95. The fully assembled and tested unit is priced at $64.95. Five different models are available. Reader Service number G29.



The RS-232-X3.

TRS-80 Software

Two S Enterprises has developed several unique programs written especially to utilize the features of the Radio Shack TRS-80. The software includes a music program, machine-language games, new variations of Lunar Lander and programs to demonstrate the graphic capabilities and formatting features of the micro.

The music program plays one of six tunes through a nearby radio by taking advantage of the logic switching noise which all computers make. This program is written to play the tunes like a jukebox until you want a new tune, then you write your own using the subroutines that produce each note. It runs in 4K, Level I and costs $13.

The Computer Art Interpreter enables the user to manipulate graphics in order to draw pictures on the screen using the program's 22 commands (e.g., UP, DOWN, HOME, etc.). When the desired shape or picture is achieved, the picture may be saved onto tape and loaded in later for display and modification. User defined shape may be called in at any time during execution of the program and placed anywhere on the screen. Cost for the 4K Level I utility package is $9.98.

The Graphic Message Printer enables the user to print out desired messages under a full graphic character set including punctuation. This Level I program allows messages up to a maximum of 12 characters in length. It sells for $7.50.

Two S Enterprises, 6851 Mammoth Ave., Van Nuys CA 91405. Reader Service number T63.

16K Static RAM Boards

Gimix, Inc., now has two versions of 16K static RAM boards for the SS-50 bus. Both use TMS 4044 RAMs, have gold bus connectors and are tested at 2 MHz. They have DIP-switch-controllable addressing, write protect and enabling of each 4K block, which allows, for example, the user to put 4K in high memory for DOS and the remaining 12K in low memory.

With the above features only, the price is $298.13. The deluxe version is socketed and GHOSTable (software-controllable readdressing, write protect and enabling of each 4K block) for $368.16.

Gimix, Inc., 1337 W. 37th Place, Chicago IL 60609. Reader Service number G28.

COMPUTER CLINIC

I have an S.D. Sales Expandoram memory board. Can any of your readers tell me how to configure the board for 56K of memory? I need the memory to go from 2000H through FFFFH, and I assume I will need to use a mixture of 4115 and 4116 memory chips.

David L. Johnson
4106 Montreal Ave.
Prince George VA 23875

I have recently been putting together a home-brew system built around the SWTP 6800 system. I want to add a ROM-based 8K BASIC. SWTP doesn't have one, and I haven't found any other source. Can you help me?

Randy Reitmeyer
614 Dow Ave.
Ocean NJ 07712

Recently I purchased an Integral Data Systems IP 125 Printer. The printer works fine. I have been successful in patching all of my TSC software. This letter was prepared using the TSC Text Editor and Text Processor.

I have been unsuccessful in getting the SWTP 8K BASIC (Version 2.3) and the printer to work together. I utilized port 3 and wire-wrapped an RS-232 interface at address 800C/800D. There is some communication.

If I perform: PRINT#3, "HELLO DOLLY" the IP 125 will successfully print HELLO DOLLY. This will be followed by READY and a prompt, # at the terminal.

However, if I perform: PRINT #3,2*2 the IP 125 will print 4, but I will never get a ready or a prompt at the terminal. It is as if it is locked up waiting for a return from subroutine.

If anyone can help in this matter, I will greatly appreciate it.

Stanley F. Lundgren
19662 Lancewood Plaza
Yorba Linda CA 92686

I am fascinated by "Onward with the COSMAC Elf!" by Jeff Duntemann (February 1979, p. 66) because inexpensive parts are available in this region for home-built systems due to the large quantities of the chips manufactured here. However, magazines and articles are hard to come by. I have been trying to obtain the *Popular Electronics* articles on the COSMAC Elf, but so far without success. I wonder if your contributors or readers have back issues, reprints, etc., so that I can purchase those articles or back issues of *Popular Electronics* (August, September 1976; March, July 1977).

Moses Huang
45-C Jalan Arnap
Singapore 10

I purchased my Apple II in October 1978. I immediately sent for their "Application Note" so that I could have my color TV converted to a monitor. I waited four months and again requested this data. To date, I have submitted two letters, waited six months and still have not received anything from Apple.

It appears that Apple is too busy selling to new customers to worry about their old ones. I would have liked to know this before I bought.

In any case, if one of your writers could do an article on this procedure, with pictures, I am sure many Apple owners would appreciate it. Here in San Antonio we have two computer shops that sell the Apple. Neither has the specs nor seems to care. After an hour on the phone, I found a few TV repair shops that would do this work if they had the specifications.

Edward S. Kleitches
7207 Camino Grove St.
San Antonio TX 78227

LETTERS

Healthy Diagnosis for Micros in Small Business

The letter from Mr. Grow (April 1979 issue, p. 19) raising some questions about the value of microcomputers in small businesses prompted me to write and describe my system, since I consider it an excellent example of the benefit these machines can provide.

I am an incorporated physician, and due to the nature of my specialty (anesthesiologist), I have no separate office and no employees. However, the IRS and the state expect my corporation to function as any other, which means paperwork, paperwork, paperwork! This amounts to a lot of work for just one employee (me)! When I was bitten by the microcomputing bug in late 1977, it quickly became apparent that this hobby had the capability to not only provide some fun but free up some of my precious time to enjoy more of that fun.

The system I eventually ended up with is a North Star Horizon II computer, Hazeltine 1510 terminal and Spinterm printer. I have written 90 percent of the software used in my corporation. This includes a word processor; check register, which prints the balances and the checks in a regular business checkbook (i.e., no expensive custom check forms are needed); payroll program, which prints out all the necessary government forms, including the W2; and billing/accounts-receivable package (handles about 30 percent of my billing).

Since at present I am the only one using my software, one could say, "Never has so much programming been done for so few." However, I enjoyed the time spent as part of my hobby, and I now spend only 10 percent of the time I used to in managing the corporate business. Last year my total expenses for an accountant were $180; that is a typical *monthly* figure for many in my situation.

The upshot of all this is that with a little imagination and know-how (obtained by reading books and magazines such as *Microcomputing*) one can do wonders with a microcomputer in small business. In the near future I hope to submit an article or two which describe in more detail some of the programs mentioned above.

David M. Dolan
So. Pasadena CA

Note to TRS-80 Owners

If you have the CTR-80 cassette, you have a problem.

When the remote circuit is opened (you push or pull the remote jack), a small voltage is sent back to the erase head . . . erasing a tiny portion of your program. To test for this, put a blank tape in your recorder and press PLAY. Push in and pull out the remote jack to start and stop the machine several times. Play the tape back; you will hear a pop each time the machine stops.

We found that the voltage was from the reverse EMF from the motor field collapsing. A small diode (1 Amp, 50 volts across the brown wire from the motor—plus to plus, minus to minus) seems to cure the problem.

If you have any questions, feel free to call: (201) 767-0555.

Dan Ellsweig, Manager
Radio Shack 2233
111 Closter Plaza
Closter NJ 07264

Termination Troubles

Several readers have questioned me about the fact that an article by Clive Bolton, appearing in the April 1979 issue of *Microcomputing*, contains several paragraphs that are virtually identical to an article written by me in the same issue; they are understandably confused as to which article is the "real McCoy." The purpose of this letter is to set the record straight.

First, my article was submitted in 1977, well in advance of Mr. Bolton's. My article was also used at that time as the basis for the instructions included in Bill Godbout's motherboard kit. Because Godbout Electronics sells a lot of motherboards, these instructions were in many people's hands prior to the publication of my article in *Microcomputing*. It is fairly evident that one of these people was Clive Bolton, since substantial portions of his article are drawn from the Godbout motherboard instructions—and, I might add, without any references, footnotes or credits.

Second, although it is frustrating and annoying to have someone pass off my hard work as his own, this is not the only problem with the article. Mr. Bolton's "active terminator" will not sink current, only source it (contrary to his claims in the text). Therefore, anyone building this circuit and expecting the same level of performance as a true active terminator (such as the one I wrote about on page 53 in the April 1979 issue) is going to be disappointed. I only hope that these people do not make a judgement on the effectiveness of active termination circuitry based on their experience with Clive Bolton's design.

It's unfortunate that space in this magazine has to be spent in cleaning up someone else's mess. I do not enjoy writing letters that criticize the abilities of a fellow author. But I strongly feel that an injustice has been done not only to me, but to the editorial staff and readership of *Microcomputing*, and that I was obliged not to let it pass by unnoticed.

Craig Anderton
Clayton CA

I wrote my article after assembling an 18-slot motherboard from California Digital. I could not locate the op amp used in the terminator, so I decided to try my hand at building a cheap one using standard, easily available hobbyist components. I came up with the 7805 version, as published.

I had never seen an article on termination in a hobbyist publication and, upon the urgings of several of my associates, I wrote the said piece. It did not come easily. I am not a writer; I spent over ten hours writing different drafts until I arrived at the final copy.

It was most unfortunate that I had read the description enclosed with the motherboard several times. I must admit it was well written and sentences stuck in my mind. As far as I was concerned, most of the knowledge was common. It never dawned on me that I had copied another's work.

As Mr. Anderton noted, the 7805 in my circuit has no provision for sinking current. It seems that, in my haste, I only examined the voltage while my computer was running; it appeared quite stable.

After receiving another letter pointing to this fault, I proposed a fix: the addition of several forward-biased diodes to limit the upper swing to 2.6 V to 3.0 V, depending on the diodes used. This should provide a decent termination, albeit not an invariant one. It does not produce a very neat circuit by any means.

I wrote the article with the very best of intentions. I am afraid, however, that I acted too swiftly and with little thought. I don't think that I've hurt anyone except perhaps those who built the circuit and were disappointed in its performance. Maybe it has created questions concerning the Godbout design; I doubt that because of the fine reputation his boards already have. Furthermore, if it has, in any way whatsoever, hurt Craig Anderton or *Microcomputing*, I am most apologetic and am at your service to help remedy the situation.

Clive Bolton
Manhasset NY

Errors and SASEs

There is an obvious error, and an error that is not so obvious, in my D/A article, which starts on page 58 of the May 1979 issue. The obvious error is in paragraph 3, page 62. The second position of S1 is actually between R2 and R3, and the third position is between R1 and R2.

The other error relates to paragraph 2, page 63. The digital voltmeter will only measure the voltage drop across a resistor as described if one end of the resistor is grounded. This is because the black lead of the voltmeter is always connected to ground. However, you can determine the voltage drop across an ungrounded resistor by measuring the voltage at one end and then at the other end. The smaller value is subtracted from the larger value and the result is used in Ohm's law as stated.

How come the article on page 74 uses a small x following a number to indicate hexadecimal and the article on page 90 uses a large X in front of a number to indicate the same thing? I always thought that hex numbers should be written as F609H. This seems to be the most common usage, and we should stick to some kind of standard to avoid confusion.

As each of my articles is published I receive many calls and letters. Most are complimentary

Contest!

Basically, April was a dreary month . . . not so for Robert Cowan, author of "A Look at TRS-80 Peripherals," "best article" for April. ● Winner of a book from the Book Nook was Alexander Vance of Cary NC. ● Winner of a lifetime subscription can be found on page 4.

and/or ask for more information. It is an indication that we are on the right track, but I wish more readers would enclose SASEs. My postage bill is becoming unbearable.

Rod Hallen
Tombstone AZ

More Formatting

I thoroughly enjoyed the "Text Formatter in BASIC" article in the May issue (p. 26). The program is running on my TRS-80 32K dual-disk system. I am using an IP225 printer interfaced through a TRS-232 printer interface.

I only made a few minor changes in the program to fit my system. I am using the up-arrow for a delimiter since I was unable to make the '80 do a backslash. The PRINTSPC(K); statement in line 1206 looked foreign to me, so I am using PRINT STRING$ (K,32); which gives the desired output. I also changed INPUT statements to LINEINPUT and PRINT to LPRINT in the write portion of the program.

To allow disk storage of text files, I replaced the tape save and load portions of the program as shown on the accompanying listing.

Perhaps these notes will be useful to other TRS-80 users.

David R. Canning
Oak Creek CO

```
700 REM
702 INPUT"NAME OF TEXT FILE";DC$
704 OPEN"I",1,DC$
705 FOR I%=0TOQ:INPUT#1,L(I%),F(I%):NEXT
706 FOR I%=1 TO Q
708 IF EOF(1) THEN 722
710 INPUT#1,L$(I%)
720 NEXT I%
722 CLOSE
724 U=L(0):UN=F(0):M=U
726 FOR I%=1TOQ:B(F(I%))=I%:NEXT:B(U)=0:B(UN)=0:F(0)=0
727 B(0)=0:L(0)=0:GOTO25
750 REM
752 INPUT"NAME OF TEXT FILE";DC$
754 L(0)=U:F(0)=UN
756 OPEN"O";1,DC$
758 FOR I%=0TOQ:PRINT#1,L(I%);F(I%);:NEXT
760 FORI%=1TOQ
764 PRINT#1,CHR$(34);L$(I%);CHR$(34);
766 NEXT I%
768 CLOSE
770 GOTO25
```

Policing Copying

Several comments in the May 1979 issue (and others) have prompted me to take keyboard in hand and reply. My first gripe is with Wayne Green's discussion of the illegality of copying Instant Software programs and "sophisticated methods for preventing the copying of programs." While I can understand his reasons, I wonder if he has considered my (the consumer's) side. I have spent considerable time figuring out how to produce backup copies of Radio Shack's Editor/Assembler and trying (unsuccessfully) to back up Microchess. What does he suggest I do if my cassette player goes on a rampage and eats $30 worth of tape in the wink of an eye? Race out and buy another tape? Maybe vendors of expensive software could consider an exchange policy whereby I could return the original (now damaged) cassette and swap it for the price of a new blank tape. Without such a policy, I am not likely to be buying many expensive programs.

Second, enough has probably been said about the Apple vs TRS-80 controversy, but I can't resist the chance to put in my two cents' worth. Specifically, whoever told Thomas Knox (Letters, May 1979, p. 23) that interfacing a TRS-80 for non-RS peripherals is "virtually impossible" doesn't know whereof he speaks. I typed this letter on a Novar Selectric, using an interface that I designed (?) myself. The ASCII-to-Selectric conversion routine is contained in an S-100 RAM board that uses circuitry on the same interface card. I certainly make no claim to be any sort of a whiz at digital design, and if I can whip up an S-100 interface, it must not be too difficult . . . certainly not impossible!

William F. Aull
Pelion SC

Fine idea, and no problem. Any Instant Software customers who manage to grind up tape in a recorder can get a replacement cassette for a $1 service charge . . . and the bum cassette. These things can happen; I have the dead cassettes to prove it.—Wayne.

There appears to be a growing problem with Apple software. Some companies selling software for the Apple are so concerned with theft of their product that they are resorting to self-modifying code and programs that modify certain key registers used by the Apple monitor. This is supposed to prevent people from listing or copying the program.

This is a shortsighted position to take. The bad part of all this is that any computer is difficult at best, and sometimes impossible, for the average home-computer owner to operate. This is particularly true with a new and unfamiliar program.

One mistake on the part of the new user can turn a $20 to $500 disk-based program into useless junk. Furthermore, the new user cannot store the program on another disk for backup or more convenient use.

We suggest you don't buy software that does any of the following:

1. Executes automatically after loading.
2. Modifies the screen memory while loading.
3. Cannot be loaded from disk using the BASIC DOS commands.
4. Cannot be unlocked using the BASIC DOS commands.
5. Cannot be listed.
6. Cannot be changed.
7. Has BASIC line numbers greater than 32000.

Finally, we recommend that you try software in the computer store before you buy the software.

Paul Lamar
Lamar Instruments
Redondo Beach CA

Wanna Drag?

My Midwest Scientific Instruments 6800 turns the No. 7 Benchmark (see "BASIC Timing Comparisons," *Kilobaud*, June 1977, p. 67) in 12.8 seconds, with equally startling results on the shorter runs. How is it possible to get this kind of performance from the old straight 8? Simple!

Start with the Software Dynamics Operating System. It's interrupt-driven, so there's no problem with timing loops on your disk drives or printer when you jack your CPU up to 2 MHz. Add Software Dynamics BASIC, and you're off.

Of course, some nitpicker will point out that SD BASIC is a compiler and that's cheating. But my experience says any time you're going to run a job more than once, you're better off with the compiler. Furthermore, SD BASIC with SDOS as implemented on the MSI 6800 is as easy and quick to use as most interpreters, effectively eliminating the major argument in their favor!

The bottom line for those who have come to believe the 6800 is all show and no go is that it has the capability to outrun the competition if you give it the right support.

Robert B. Peirce
McMurray PA

BOOKS

(from page 14)

explain it to me, but the tunnel from my listening ear through to the exit ear was functioning well at that moment. Some of the other statements, while not peculiar to North Star BASIC, could stand further treatment. What, for example, does "OUT 5,3" mean? ASC, VAL, CALL and INP, Boolean operators and string arrays need discussion. These, I expect, will be in the second edition.

Paul W. Marsh
Bozeman MT

F. R. Ruckdeschel
773 John Glenn Blvd.
Webster NY 14580

IC Logic Tester and Parallel I/O Expander

Use this to expand the number of TTL-level parallel lines into and out of an 8-bit port.

The use of microcomputers to test integrated circuits is a natural and simple application that has been referred to in several small-system journals over the past few years. However, so far no one has published an article that provides not only the philosophy for the device structure, but also a complete description of the hardware configurations as well as the required test software. The following discussion, I hope, will fill this gap in the literature.

Design Goals

Prior to embarking upon this exercise, I will jot down a few basic operational characteristics that I established as design goals.

1. The device was to be capable of testing a wide, though not necessarily complete, class of common integrated circuits.

2. The test software was to be diagnostic in nature. That is, the test output was not to be yes/no but rather a brief description of the fault.

3. For the bargain hunters among us who sometimes obtain uncoded or cryptically labeled chips, the apparatus was to be capable of determining the test IC device number over a limited set.

4. The test hardware was also not to be limited to IC testing only but capable of other important functions, such as parallel I/O port expansion and in-circuit IC evaluation.

In response to the first goal, the design to be shown can accommodate TTL-level integrated circuits having 14- or 16-pin DIP configurations in which the 5 volt dc power is applied to the last pin (pin 14 for 14-pin devices; pin 16 for 16-pin devices) and ground applies to the middle number pin (pin 7 or 8, respectively). This covers a lot of circuits within the SN7400 series, as well as many other devices such as the 8T97 Tristate gate, etc.

The second goal was easily achieved, but at the expense of lengthy software. However, the running time for examining a specific IC is relatively short and, including new tests, is easily achieved by adding the appropriate subroutines.

As an extension of the above use, you can also perform in situ tests of integrated circuits. This is particularly helpful when an external circuit is not operating properly and the component ICs are soldered in. In this case a ribbon cable connector can be run from the test socket to a clip on the in-circuit IC to be tested, and the signals within the actual external circuit can be employed to direct the examination. This application will be discussed in greater detail later.

The third goal, to sort out unmarked or mysteriously coded integrated circuits, is achieved with little additional software. It uses the specific test routines employed to check out a particular chip and goes through the available repertoire of tests until a successful input/output scan has been encountered. The corresponding device number is then printed out, along with other pertinent data.

This operation is often lengthy because a large part of the test library may have to be exercised before a match is found. It is intriguing to place a known IC into the test socket and have the computer guess its number! However, as pull-up resistors are used in the test circuit, open collector outputs cannot be distinguished from TTL-level outputs.

The fourth goal, having alternate functions, was also intentionally achieved by designing the test apparatus with the objective of also using it to expand the number of bits that could be communicated through an 8-bit parallel I/O port. This was accomplished at the expense of some software overhead and a loss in port operating speed when the option was actually used. However, few common real-time operations (e.g., digital voltmeter data gathering) require data acquisition rates that would find this limitation objectionable.

After completing the design, and, admittedly, as an afterthought, I noticed that the test apparatus could work in reverse. That is, the tester could also perform the logical functions of an integrated circuit as defined by the computer. This can be achieved by running a two-ended (male) DIP ribbon connector from the test socket to the socket where an integrated circuit is to be "simulated."

The chief projected drawbacks of this are that the simulated device speed is slow and the fanout of the simulated IC low. There is also the minor consideration that one IC is replaced by an entire computer!

Integrated circuits are available through Digi-Key Corporation, PO Box 677, Thief River Falls MN 56701, Tel. No. (218) 681-6674. Negatives for the double-sided printed circuit board are available through F. Ruckdeschel, 773 John Glenn Blvd., Webster NY 14580. The software driver is also available from the author in the following forms:

ASCII Listing:	Paper Tape $10
	Cassette (Standard Modem frequencies 2025/2225 Hz) $8
	Cassette (Tarbell format) $8.
Machine Code:	North Star Diskette $10
	Cassette (Mits 8K BASIC Version 3.2; 88-ACR) $6
	Cassette (Mits 12K BASIC Version 3.2; 88-ACR) $6
	Cassette (Mits 12K BASIC Version 4.0; 88-ACR)

Part	Quantity	Description	Price (each)
ICs A,B	2	7404 hex inverters	$0.21
ICs C,D,E,F,G,H,I,J	8	7400 Quad NAND gates	0.21
ICs K,L,M,N	4	74279 Quadruple $\overline{S}$-$\overline{R}$ latches	0.55
ICs O,P,Q,R	4	74365 hex bus drivers	0.67
Capacitors	27	0.1 uF, 50 volts	about $0.10
Capacitor	1	0.01 uF, 50 volts	about $0.10
Capacitor	1	Electrolytic 100 uF/16 volts	about $0.20
Resistors	14	330Ω, 1/4 Watt	about $0.02-0.10
Resistors	16	1000Ω, 1/4 Watt	about $0.02-0.10
IC Sockets	2	14- and 16-pin sockets	about $0.20 ea
Printed circuit board negatives	1	Double-sided; undrilled	$15 Available from author

Table 1. Parts list.

However, those who have their own computers are not necessarily concerned with practicality anyway.

In the ensuing sections the basic device structure is discussed, highlighted by a schematic and an attendant circuit description. The printed circuit foil patterns are then presented, followed by an outline of the driving software for testing integrated circuits. A partial software listing is also provided.

Subsequently, the port expansion application is considered, as well as the in-circuit IC test operation. In the latter application it is proposed that *two* integrated circuit testers be used—one to probe the in-circuit IC and one to compare those results with the response of a corresponding known-good IC. The software overhead associated with this comparison approach is minimal.

Device Structure

Assuming that the power and ground requirements for the IC to be examined are provided by hard-wire connections on the tester printed circuit board (as they are in the foil layout patterns to be presented), 14 bits are required to probe a 16-pin 7400 series chip. This is consistent with using the eighth bit in a computer output byte to direct the remaining seven bits into one of two channels.

Thus an 8-bit parallel port can be used to output the required 14 bits of information in two groups of seven. The eighth bit in the computer output byte can also be used to multiplex the input lines; two 8-bit words may be input through a single 8-bit port.

Thus the expansion/compression possible is: eight bits out expanded to 14 bits out and 16 bits in compressed to eight bits in.

The general circuit philosophy for accomplishing this is simple. The seven data bits coming into the tester board are presented to two 7-bit sets of latches. The eighth data bit determines which latch set is enabled. The 14 outputs (Q_0) of these latches are the desired expansion. The 16 bits to be sent to the computer are presented to two 8-bit sets of Tri-states, each set being enabled by the above eighth bit control signal and each set connected in parallel. Moving the data around is thus under software control by the computer via the eighth data bit.

The schematic of a circuit that accomplishes the task is shown in Fig. 1. The integrated circuit components used were chosen for their low cost, ready availability and reliability. Their total cost is less than $6 (see Table 1). A more efficient circuit design is possible using 74100 or CMOS latches, but those chips and their accompanying sockets are, on the whole, more expensive.

The input data byte (it is assumed that the eighth bit of the computer output port is latched) is filtered before presenting it to the 7404 inverters, ICs A and B. This removes line noise as well as creates a lag in the transition of the seven data bits rel-

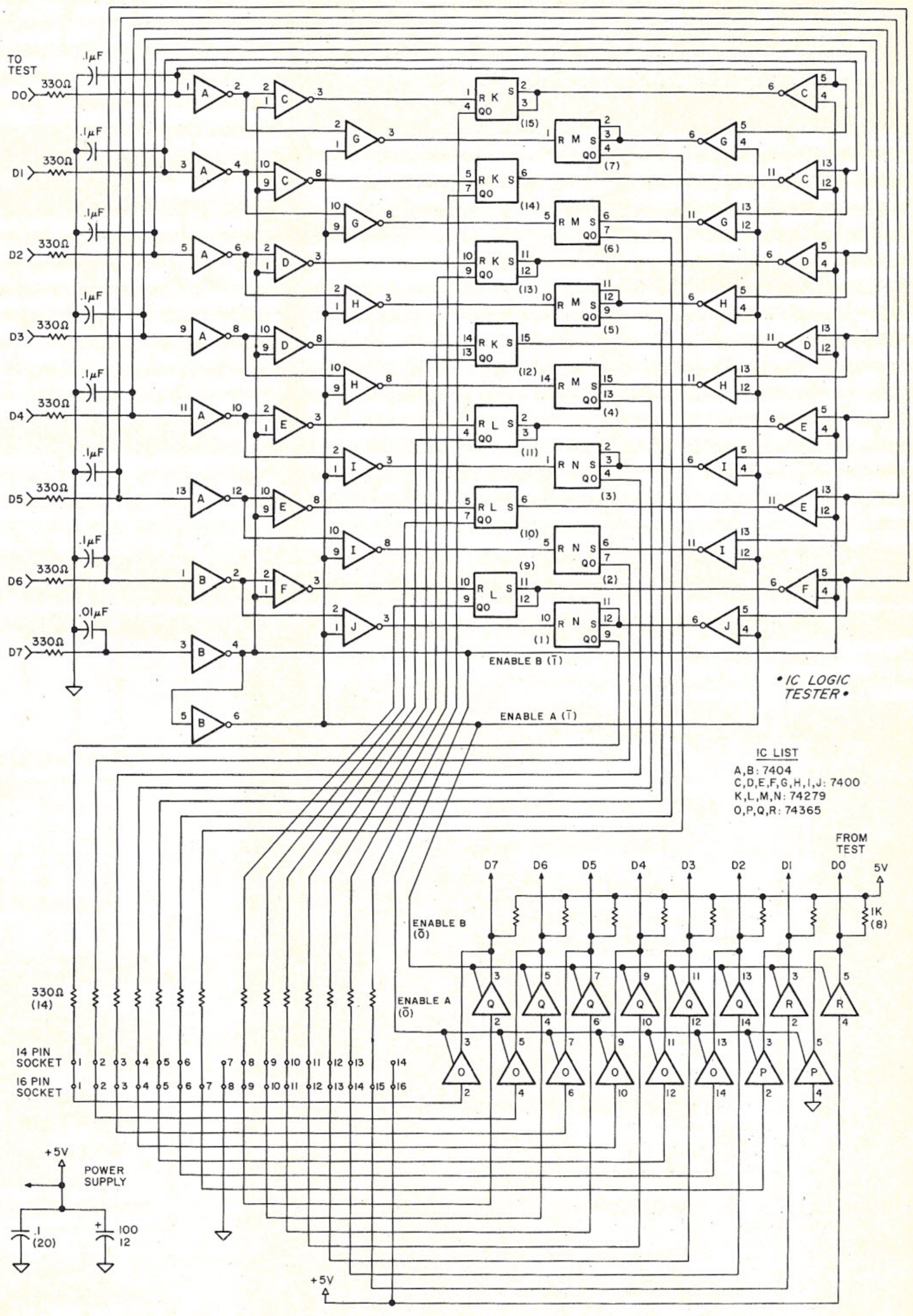


Fig. 1. Schematic of integrated circuit logic tester. The schematic was purposely laid out symmetrically to aid in the original wire-wrapping. One more IC (a 7400) than necessary is used. Note that each IC (and test socket) contains a bypass capacitor—20 in all. By changing the capacitors at the tester input, the device test speed may be increased. However, this is not a concern when the driver software is written in BASIC as interpreters are slow.

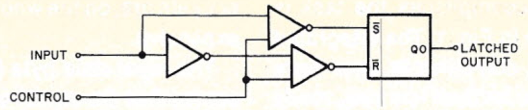


Fig. 2. BASIC controlled latch.

Inputs		Output
R	S	Q (pseudo-stable)
1	1	Q_0
0	1	1
1	0	0
0	0	1 (pseudo-stable)

Table 2. $\overline{R}$-$\overline{S}$ latch truth table (double S inputs connected).

ative to the transition of the control bit; it is not good practice to have the data bits changing while the multiplex channels are simultaneously being switched.

This lag is created by having 0.1 uF capacitors on the seven data lines and a 0.01 uF capacitor on the control line. The resulting RC time constants are 100 and 10 microseconds, respectively. The same effect could be accomplished in software by outputting a change in the control bit (keeping the other seven bits constant), later followed by the seven new data bits to be put into the other multiplex channel.

The 7404 inverters are partly used as inexpensive signal conditioners and drivers; the drain on the computer output port is one TTL load. They also provide the inverted signals, which are later needed for loading the latches. The control bit is reinverted such that the control of each channel may be subsequently performed symmetrically, with the asymmetry vital to the multiplexing being right at the control line outputs of the 7404 inverter, B.

Thus the schematics for each of the two input and output channels are nearly identical beyond this point. If something goes wrong with the tester itself, you can compare corresponding nodes to isolate the flaw.

The NAND gates (ICs C through J) and the $\overline{R}$-$\overline{S}$ latches (ICs K through N) form the heart of the switched latch structure. The truth table for the $\overline{R}$-$\overline{S}$ latch is shown in Table 2.

Logic Features

The key logic features to notice are: (1) Q_0, the "remembered" or latched output is maintained by holding both the $\overline{R}$ and $\overline{S}$ inputs high. (2) The value of Q_0 can be controlled by previously having either a (0,1) or (1,0) combination on the $\overline{R}$-$\overline{S}$ inputs.

To load the latch we require the bit destined to be saved to be placed at the $\overline{S}$ input and its inverse to be placed at $\overline{R}$. To then latch it, both $\overline{R}$ and $\overline{S}$ should be driven high based on the control bit logic level. If the $\overline{R}$ and $\overline{S}$ inputs had been previously NANDed with the control bit (active low to latch), this sequence is accomplished. The basic circuit that provides this function is shown in Fig. 2 (Not an original circuit. It appeared in the September 1977 (Microelectronics) issue of *Scientific American*, for example). The input from the computer port is sampled by placing a 1 on the particular control line (there are two control lines which stem from the 7404 inverter, IC B) and latched with a 0.

This circuit is replicated in two sets of seven. The only difference between the two sets is that one is sampling while the other is latched and vice versa. The two control lines that accomplish this are labeled "Enable A" and "Enable B."

The 14 latched outputs are presented to the test socket pins through 330Ω resistors. If the IC in the test socket has a legitimate input at a particular

Program listing. Partial software listing of the IC tester program. It is written in North Star BASIC, Version 6, Release 3. The statements were adjusted for the limited test set.

```
10 REM SN7400 SERIES IC TESTER
20 REM F.R. RUCKDESCHEL  6/3/78
30 REM O=OUTPUT TO AND I=INPUT FROM IC TESTER
40 REM J1=RANGE OF SERIES TEST IN THIS PROGRAM
50 J1=7410
60 DIM O(17),I(17),A$(11),C$(27),O1(17),U$(14)
70 GOSUB 1920
80 B$="ERROR"
90 C$="TEST COMPLETED SUCCESSFULLY"
100 U$="UNKNOWN IC IS "
110 GOSUB 1830
120 PRINT"***INSTRUCTIONS***"
130 GOSUB 1830
140 PRINT"TURN TESTER POWER OFF"
150 GOSUB 1870
160 PRINT"INSERT IC TO BE TESTED"
170 PRINT"TURN POWER ON"
180 GOSUB 1870
190 GOSUB 1520
200 IF I(8)=1 THEN GOTO 220
210 IF I(16)=1 THEN GOTO 300
220 PRINT B$
230 FOR J=1 TO 16
240 PRINT I(J),
250 NEXT J
260 PRINT
270 PRINT"CHECK EITHER CABLE OR POWER SUPPLY"
280 PRINT "READY?"\INPUT M$
290 GOTO 140
300 GOSUB 1830
310 P9=1
320 PRINT "SEARCH OR TEST (S/T)",
330 INPUT M$
340 IF M$="S" THEN GOTO 600
350 IF M$="T" THEN GOTO 370
360 GOTO 330
370 PRINT"INPUT THE DEVICE NUMBER: ",
380 INPUT S
390 P9=0
400 REM P9=0 MEANS TEST
410 REM P9=1 MEANS SEARCH
420 REM BRANCH TO TEST SUBROUTINES
430 IF S<7400 THEN GOTO 370
440 IF S=8000 THEN GOSUB 1960
450 IF S=8000 THEN GOTO 130
460 IF S>J1 THEN GOTO 130
470 IF S=7400 THEN GOSUB 1120
480 IF S=7401 THEN GOSUB 2300
490 IF S=7402 THEN GOSUB 2430
500 IF S=7403 THEN GOSUB 2720
510 IF S=7404 THEN GOSUB 2790
520 IF S=7405 THEN GOSUB 3060
530 IF S=7406 THEN GOSUB 3130
540 IF S=7407 THEN GOSUB 3200
550 IF S=7408 THEN GOSUB 3470
560 IF S=7409 THEN GOSUB 3760
570 IF S=7410 THEN GOSUB 3830
580 GOTO 130
590 REM **********************
600 REM SEARCH EXECUTIVE
610 S9=0\GOSUB 1180\IF S9=0 THEN GOSUB 1120\IF S9=0 THEN STOP
620 S9=0\GOSUB 2360\IF S9=0 THEN GOSUB 2300\IF S9=0 THEN STOP
630 S9=0\GOSUB 2470\IF S9=0 THEN GOSUB 2430\IF S9=0 THEN STOP
640 S9=0\GOSUB 2820\IF S9=0 THEN GOSUB 2790\IF S9=0 THEN STOP
650 S9=0\GOSUB 3230\IF S9=0 THEN GOSUB 3200\IF S9=0 THEN STOP
660 S9=0\GOSUB 3510\IF S9=0 THEN GOSUB 3470\IF S9=0 THEN STOP
670 S9=0\GOSUB 3880\IF S9=0 THEN GOSUB 3830\IF S9=0 THEN STOP
680 GOSUB 4140
690 REM *********************
700 REM ENCODER AND OUTPUT ROUTINE
710 REM ***FOR 16 PIN IC***
720 O1=128
730 T=1
740 FOR I=7 TO 1 STEP -1
750 O1=O1+O(I)*T
760 T=2*T
770 NEXT I
780 O2=0
790 T=1
800 FOR I=15 TO 9 STEP -1
810 O2=O2+O(I)*T
820 T=2*T
830 NEXT I
840 REM OUTPUT SIGNALS
850 OUT 39,O1
860 OUT 39,(O1-128)
870 OUT 39,O2
880 REM SIGNALS NOW ON PINS
890 RETURN
900 REM *********************
910 REM ***ENCODER ROUTINE***
920 REM ***FOR 14 PIN IC***
930 FOR I=1 TO 16
940 O1(I)=O(I)
950 NEXT I
960 FOR I=14 TO 8 STEP -1
970 O(I+1)=O(I)
980 NEXT I
990 O(8)=1
1000 GOSUB 700
1010 REM SIGNALS ARE NOW ON PINS
1020 RETURN
1030 REM *********************
1040 REM ***DECODER ROUTINE***
1050 REM ***FOR 14 PIN IC***
```

pin, the load through the corresponding resistor will be 1.6 mA or less. At most this causes a half volt drop across the 330Ω series resistor; that pin will therefore be controlled by the output of the corresponding latch. If instead the pin at the test socket corresponds to an output on the test IC and if this output can sink at least 6 mA, then a 0 at that output pin will not be changed by any latch output on the other side of the 330Ω resistor.

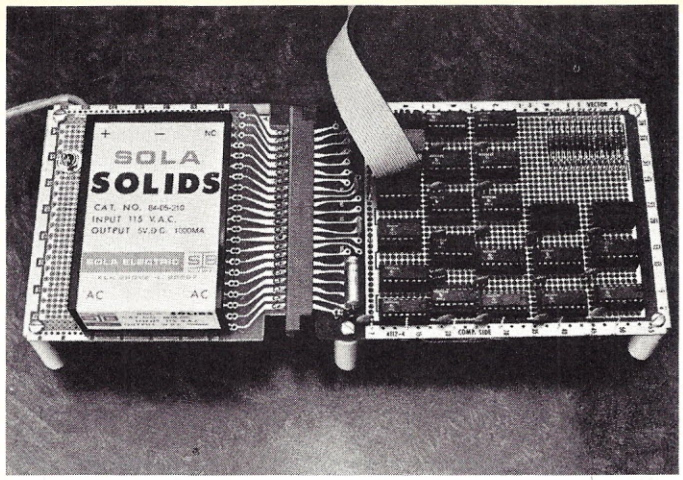
The current-sinking capabilities of 7400 (16 mA), 74S00 (20 mA), 74H00 (20 mA) and 74LS00 (8 mA) series integrated circuits fall in the allowable range. The 74L00 (3.6 mA) series may have some current-sinking difficulties, though the samples tested thus far have not demonstrated this problem. Similarly, if any test IC pin is shorted to either ground or Vcc, the corresponding logic state is unaffected by the latched logic level.

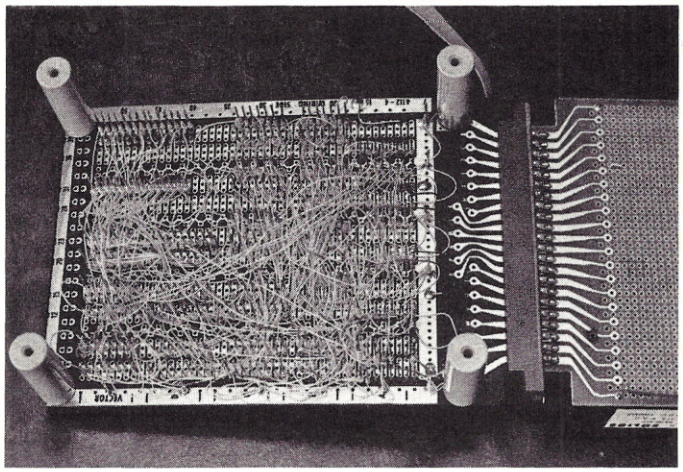
The remaining possibility (other than an intermediate state) is that the pin being considered is an output, is at a logic 1 level and does not have the capability to support several milliamperes' current drain. In that case a 0 on the latch may affect (load down) the legitimate output.

Thus, in an IC test, the logic levels at the latch locations corresponding to test IC outputs should be at a logic 1 state unless a short to +Vcc is being tested against. In the software driver routine presented later, the logic levels at the latches are kept at a logic level 1 unless particular inputs are being controlled.

The 16 logic levels at the test socket include ground and Vcc; they allow a check of whether or not the tester board is correctly powered upon initial start-up. These levels are multiplexed into two bytes by Tri-state buffers (ICs O, P, Q and R) having pull-up resistors on their outputs. While one set of eight bits is enabled by the control line, the other eight are disabled and vice versa.

This completes the discus-





Photos 1a and 1b. Note that the power supply is mounted on a separate board that simply plugs into the wire-wrap prototype. This is done as routine practice so that power supply hookup is fast and simple; one supply can be used for several projects. Current drain in this case was 400 mA (maximum).

```
1060 GOSUB 1520
1070 FOR I=8 TO 14
1080 I(I)=I(I+1)
1090 NEXT I
1100 RETURN
1110 REM ********************
1120 REM***7400 TEST ROUTINE***
1130 GOSUB 1830
1140 PRINT"*** 7400 ***"
1150 GOSUB 1830
1160 PRINT"QUADRUPLE 2-INPUT"
1170 PRINT"POSITIVE-NAND GATES"
1180 REM UNKNOWN ENTRY POINT
1190 A=1\B=2\C=3\GOSUB 1260
1200 A=4\B=5\C=6\GOSUB 1260
1210 A=9\B=10\C=8\GOSUB 1260
1220 A=12\B=13\C=11\GOSUB 1260
1230 IF P9=0 THEN PRINT C$\RETURN
1240 REM ********************
1250 REM TWO INPUT NAND GATE TEST (14 PIN)
1260 FOR X1=0 TO 1
1270 FOR Y1=0 TO 1
1280 REM SEARCH*PREVIOUS ERROR=SKIP
1290 IF P9*S9=1 THEN GOTO 1480
1300 REM INITIALIZATION
1310 GOSUB 1750
1320 O(A)=X1\O(B)=Y1
1330 REM OUTPUT TO 14 PIN ENCODER ROUTINE
1340 GOSUB 910
1350 REM INPUT SIGNALS TO COMPUTER
1360 GOSUB 1520
1370 REM GOTO 14 PIN IC DECODER
1380 GOSUB 1040
1390 IF NOT(O1(A) AND O1(B))=I(C) THEN GOTO 1480
1400 S9=1
1410 REM IF SEARCH THEN SKIP PRINTOUT
1420 IF P9=1 THEN GOTO 1480
1430 PRINT B$
1440 PRINT O1(A)," (PIN ",A,") NAND ",O1(B)," (PIN ",B,") EQUALS ",
1450 PRINT NOT(O1(A) AND O1(B))," , NOT ",I(C), "(PIN ",C,")"
1460 STOP
1470 GOSUB 1830
1480 NEXT Y1
1490 NEXT X1
1500 RETURN
1510 REM ********************
1520 REM 16 PIN IC INPUT AND DECODER ROUTINE
1530 IF O2>127 THEN O2=O2-128
1540 IF O1>127 THEN O1=O1-128
1550 OUT 39,128+O1
1560 I1=INP(37)
1570 OUT 39,O2
1580 I2=INP(37)
1590 T=256
1600 FOR I=0 TO 7
1610 T=T/2
1620 IF I2-T <0 THEN GOTO 1660
1630 I2=I2-T
1640 I(9+I)=1
1650 GOTO 1670
1660 I(9+I)=0
1670 IF I1-T< 0 THEN GOTO 1710
1680 I1=I1-T
1690 I(I+1)=1
1700 GOTO 1720
1710 I(I+1)=0
1720 NEXT I
1730 RETURN
1740 REM ********************
1750 REM ***INITIALIZATION***
1760 FOR I=1 TO 16
1770 I(I)=1
1780 O(I)=1
1790 NEXT I
1800 RETURN
1810 REM ********************
1820 REM ***TRIPLE CARRIAGE RETURN***
1830 PRINT\PRINT\PRINT
1840 RETURN
1850 REM ********************
1860 REM READY PROMPT
1870 PRINT "READY",
1880 INPUT M$
1890 REM ********************
1900 REM ***PORT INITIALIZATION***
1910 REM  37=INPUT PORT    39=OUTPUT PORT
1920 OUT 36,0\OUT 38,0\OUT 37,0
1930 OUT 39,255\OUT 36,44\OUT 38,44
1940 RETURN
1950 REM ********************
1960 REM BOARD TEST ROUTINE
1970 PRINT "IC TESTER CHECKOUT ROUTINE"
1980 PRINT "REMOVE IC FROM TEST SOCKET"
1990 GOSUB 1870
2000 PRINT
2010 Q=0
2020 N=0
2030 FOR M=0 TO 1
2040 FOR J=1+N TO 7+N
2050 GOSUB 1750
2060 REM OUTPUTS ARE INITIALIZED
2070 O(J)=M
```

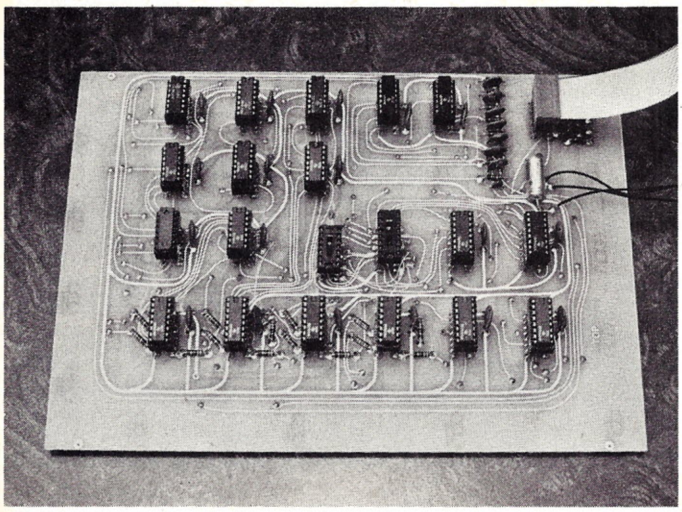


Photo 2. The (near) final version of the etched pattern tester. Power supply leads are shown. Note that the ground lead of the power supply is also a signal ground and should be connected to the signal ground of the parallel I/O port.

sion of the tester circuit schematic.

The prototype circuit was originally wire-wrapped together and is shown in Photos 1a and 1b. The working circuit was then transferred to the double-sided foil patterns shown on Figs. 3a and 3b. A photograph of a populated etched pattern board is also shown. A few errors occurred in this version, but they have been corrected on the foil patterns presented. Note that the sockets are necessary only for the two test locations (14 and 16 pin).

Providing DIP sockets only at the test locations not only keeps down the project cost, but is also conducive to using the IC component pins themselves as feed-throughs. Unfortunately, I was not ingenious enough to avoid having to use quite a few jumpers. However, at least they are clearly labeled.

Test Software

The test software consists largely of placing known signals on the test IC inputs and examining whether or not the resulting outputs are in accordance with the truth table for that device. Although the partial listing of the BASIC program accompanying this article is very long, it is also very simple.

A brief discussion of the data bit input/output operation of this program is appropriate, particularly as the principles are relevant to a later consideration of operating the tester as a parallel port expander/compressor. The complete program, which is available from me, tests more than 100 integrated circuits.

We first consider placing signals on the 16-pin test socket. The correlation with the pins on the 14-pin socket is apparent from the schematic.

If $D_7 = 0$, the seven input bits are placed (but not latched) onto pins 9 through 15 according to Table 3. The control line also selects which eight test socket pin logic levels are sent *from* the board to the computer port. The code is shown in Table 4.

If no IC is placed in the test socket, then the latches (ICs K, L, M and N) will control the logic levels at the test socket pins, except for the power supply and ground, which can be disconnected. In that case, if DO designates data bits into the tester—*out* of the computer—and DI designates data bits out of the tester—*into* the computer—(S-100 convention), then the input/output characteris-

```
2080 GOSUB 700
2090 REM SIGNALS PLACED ON SOCKET PINS
2100 GOSUB 1520
2110 REM SIGNALS RECOVERED FROM TEST SOCKET
2120 REM COMPARE INPUT WITH OUTPUT
2130 IF Q>0 THEN GOTO 2210
2140 IF I(16)<1 THEN GOTO 2170
2150 IF I(8)=0 THEN GOTO 2210
2160 PRINT
2170 PRINT"POWER SUPPLY ERROR"
2180 PRINT"PIN 8=",I(8),"    PIN 16=",I(16)
2190 Q=1
2200 PRINT
2210 IF O(J)=I(J) THEN GOTO 2240
2220 PRINT "ERROR AT PIN ",J,",OUTPUT=",O(J)," INPUT=",I(J)
2230 STOP
2240 NEXT J
2250 NEXT M
2260 N=N+8
2270 IF N<8 THEN GOTO 2030
2280 PRINT C$\RETURN
2290 REM ******************
2300 REM 7401 TEST ROUTINE
2310 PRINT "*** 7401 ***"
2320 GOSUB 1830
2330 PRINT "QUADRUPLE 2-INPUT"
2340 PRINT "POSITIVE-NAND GATES"
2350 PRINT "WITH OPEN COLLECTOR OUTPUTS"
2360 REM UNKNOWN ENTRY POINT
2370 A=2\B=3\C=1\GOSUB 1260
2380 A=5\B=6\C=4\GOSUB 1260
2390 A=8\B=9\C=10\GOSUB 1260
2400 A=11\B=12\C=13\GOSUB 1260
2410 IF P9=0 THEN PRINT C$\RETURN
2420 REM *******************
2430 REM 7402 TEST ROUTINE
2440 PRINT "*** 7402 ***"
2450 PRINT "QUADRUPLE 2-INPUT"
2460 PRINT "POSITIVE NOR GATES"
2470 REM UNKNOWN ENTRY POINT
2480 A=2\B=3\C=1\GOSUB 2540
2490 A=6\B=5\C=4\GOSUB 2540
2500 A=8\B=9\C=10\GOSUB 2540
2510 A=11\B=12\C=13\GOSUB 2540
2520 IF P9=0 THEN PRINT C$\RETURN
2530 REM *******************
2540 REM TWO INPUT NOR GATE TEST (14 PIN)
2550 FOR X1=0 TO 1
2560 FOR Y1=0 TO 1
2570 IF S9*P9=1 THEN GOTO 2680
2580 GOSUB 1750
2590 O(A)=X1\O(B)=Y1
2600 GOSUB 4190
2610 IF NOT(O1(A) OR O1(B))=I(C) THEN GOTO 2680
2620 S9=1
2630 IF P9=1 THEN GOTO 2680
2640 PRINT B$
2650 PRINT O1(A)," (PIN ",A,") ",
2660 PRINT "NOR ",O1(B)," (PIN ",B,") EQUALS ",NOT (O1(A) OR O1(B))," NOT ",I(C)," (PIN ",C,")"
2670 STOP
2680 NEXT Y1
2690 NEXT X1
2700 RETURN
2710 REM ********************
2720 REM 7403 TEST ROUTINE
2730 PRINT "*** 7403 ***"
2740 PRINT "QUADRUPLE 2-INPUT"
2750 PRINT "POSITIVE-NAND GATES"
2760 PRINT "WITH OPEN COLLECTOR OUTPUTS"
2770 GOTO 1190
2780 REM ********************
2790 REM 7404 TEST ROUTINE
2800 PRINT "*** 7404 ***"
2810 PRINT "HEX INVERTERS"
2820 REM UNKNOWN ENTRY POINT
2830 A=1\B=2\GOSUB 2910
2840 A=3\B=4\GOSUB 2910
2850 A=5\B=6\GOSUB 2910
2860 A=9\B=8\GOSUB 2910
2870 A=11\B=10\GOSUB 2910
2880 A=13\B=12\GOSUB 2910
2890 IF P9=0 THEN PRINT C$\RETURN
2900 REM ********************
2910 REM INVERTER TEST (14 PIN)
2920 FOR X1=0 TO 1
2930 IF P9*S9=1 THEN GOTO 3030
```

ITHACA AUDIO

THE OEM MARKETPLACE

Assembled and Tested Added at Ithaca Audio

Field-proven reliable engineering

Over 15,000 boards worldwide prove Ithaca Audio provides the quality and reliability you demand.

Ithaca Audio Boards are fully S-100 compatible, featuring gold edge connectors and plated-through holes. All boards (except the Protoboard) have fully buffered data and address lines, DIP switch addressing, solder mask and parts legend.

- **Z-80 CPU Board** still the most powerful 8 bit central processor available. Featuring power-on-jump, provision for on-board 2708. Accepts most 8080 software.

A&T 4 mHz	**$205.00**
A&T 2 mHz	**$175.00**
Blank PC	**$ 35.00**

- **Disk Controller Board** controls up to 4 single or double sided drives. Supported by a host of reliable software packages: K2 FDOS, Pascal, Basic and complete diagnostics.

A&T	**$175.00**
Blank PC	**$ 35.00**

- **K2 FDOS** Disk software in the DEC tradition. Includes character oriented text editor (TED), File Package (PIP), Debugger (HDT), Assembler (ASMBLE), HEXBIN, 1 COPY, System Generator (SYSGEN) and more. Command syntax follows Digital's OS-8/RT-11 format. First in a family of high level software. Basic and Pascal available now. Soon-to-be-released Fortran.

K2 Disk	**$ 75.00**

- **Video Display Board** features the full 128 upper/lower case ASCII character set. Easy-to-read 16 line x 64 character format can be displayed on an inexpensive video monitor or modified TV set. Includes TTY software. Add our powerful K2 FDOS to create a versatile operator's console.

A&T	**$145.00**
Blank PC	**$ 25.00**

- **8K Static RAM Board** High speed static memory at a reasonable cost per bit. Includes memory protect/unprotect and selectable wait states.

A&T 250 ns	**$195.00**
A&T 450 ns	**$165.00**
Blank PC	**$ 25.00**

- **2708/2716 EPROM Board** Indispensable for storing dedicated programs and often used software. Accept up to 16K of 2708's or 32K of 2716's.

A&T (less EPROMs)	**$ 95.00**
Blank PC	**$ 25.00**
2708 EPROMs	**$ 11.00**

The leading manufacturer of blank S-100 boards is adding a new wrinkle—now all their boards are available assembled and tested. "This is a natural progression for the company" according to Mr. James Watson, President. "Actually we've been supplying assembled and tested for some time to our volume customers and OEM's, particularly those overseas. Our production staff is now fully up to speed, so just about everything is available from stock." The company scheduled 6 months to phase in assembled and tested to allow time to build base inventories, before offering the boards to the public. "We feel this is quite important. A lot of companies have earned themselves a bad name in this business by announcing products they can't really deliver. We simply won't do that." Mr. Watson further explained that Ithaca Audio intends to remain leader in blank boards and expects to release a minimum of 6 new designs by August, which will be offered both blank and assembled and tested.

Memory Prices Tumble

Ithaca Audio first to break 1¢/Byte Barrier

By cutting prices for 32K of RAM to $319 Ithaca Audio becomes the first computer vendor ever to offer high speed memory for less than a penny a byte. Commenting on the announcement, Steve Edelman, Director of Engineering said "Just a few years ago people were wishing for a penny a *bit*, and even now memory for most large computers costs about 2¢/byte and that's only in 1 Megabyte chunks." In fact it's the relative modest capacity of the 32K board that makes it so interesting. Users need not buy the full 64K to take advantage of the low price per bit. Furthermore, the board is available both as a kit and assembled and tested.

Delivery is stock to two weeks. Pricing is:

- 32K kit **$319**
- 32K A&T **$359**
- 64K kit **$645**
- 64K A&T **$695**

8" Disk Drives

Shugart compatible Memorex 550's are in stock.

Single and double density compatible, 330K bytes capacity with our controller or use your own.

Either way **$456**

- **Protoboard** Universal wire-wrap board for developing custom circuitry. Room for three regulators. Accepts any size DIP socket.

Blank PC	**$ 25.00**

Pascal/Z Ready

The first Pascal Compiler for the Z80, and the fastest Z80 Pascal ever is now ready. Over one year in development, Ithaca Audio was obviously pleased with the results. "We really have outperformed them" states Jeff Moskow, Director of Software Engineering, beaming over the recently released benchmarks, in which Pascal/Z averaged better than five times the speed of a recent P-code implementation.

"Pseudo-code means a vendor only has to supply one compiler to lots of people using lots of different machines, and that makes his life very easy, but it also means users' programs execute significantly slower. Therefore, we chose to write a native compiler that delivers fast re-entrant ROMable code, with no need for an intermediate language and interpreter. That's where our speed comes from." As a matter of fact, Pascal/Z is often twenty times as fast as UCSD's implementation and may well be faster than dedicated Pascal machines such as the recently announced Western Digital Pascal Microengine.™ Unlike the Microengine, Pascal/Z does not require any new special CPU hardware and has the added benefit of compatibility with existing Z80 software.

Operational requirements of Pascal/Z are the Ithaca Audio K2 Operating system and 48K of memory during compiles. The output is standard Z80 Macrocode which is linked and run through the Ithaca Audio Macroassembler. Binary files may be as small as 2.5K, or even less if the full library is not used. The compiler, including the Macroassembler, is available on an 8" K2 floppy disk. Price including full documentation is $175.00. The Macroassembler is available separately for $50.00. Delivery is from stock.

More Software:

For those that don't require the speed of a compiler like Pascal/Z, Ithaca Audio also offers the convenience of BASIC. BASIC/Z, an extended version of TDL's Super Basic, runs in slightly over 12K and is supplied on an 8" K2 disk for $75.00.

SAVE Even More -

When you buy your software as a package

K2 and Pascal/Z	**$225**
	SAVE $25
K2, Pascal/Z and Basic/Z	**$275**
	SAVE $50

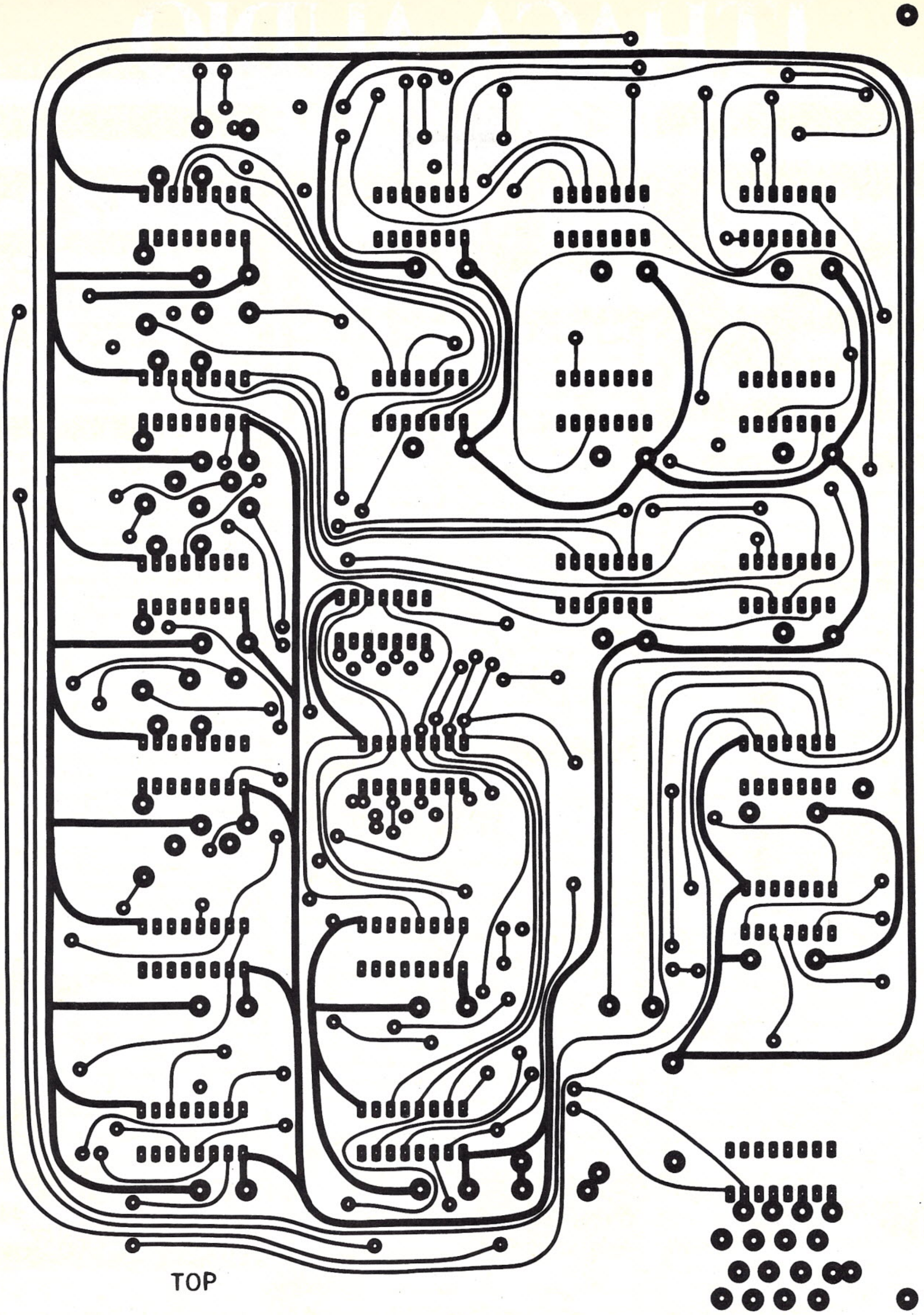
HOW TO ORDER

Send check or money order, include $2.00 snipping per order. N.Y.S. Residents include tax.

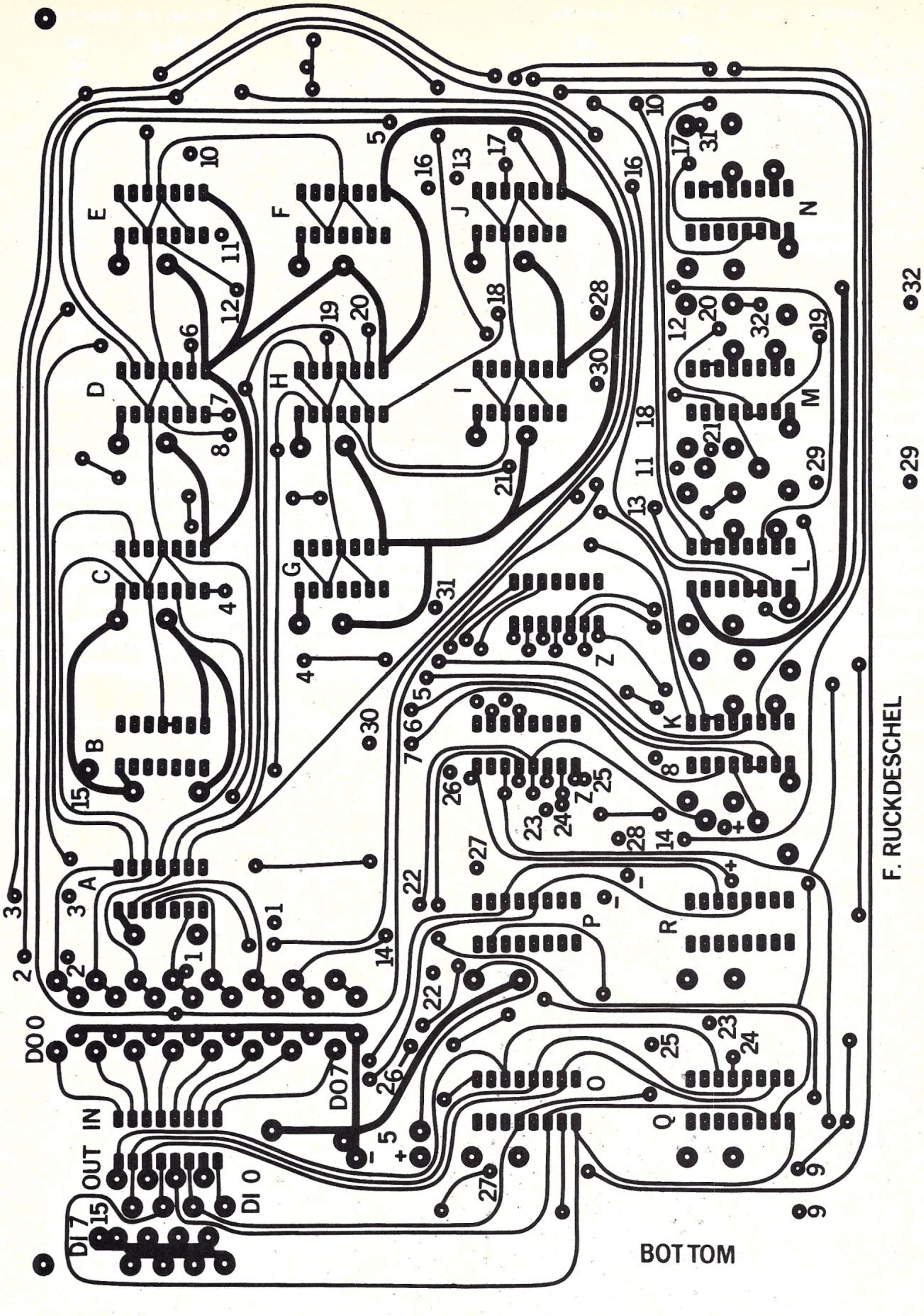
For technical assistance call or write to:

ITHACA AUDIO

P.O. Box 91
Ithaca, New York 14850
Phone: 607/257-0190



Figs. 3a and 3b. Foil patterns for the integrated circuit logic tester. The two accompanying foil patterns represent the layout for a double-sided board implementation of the IC tester schematic. Since most experimenters do not have the capability to create boards with plated-through holes, some feed-through soldering is required. Also, several jumpers must be installed. They are numbered 1 through 32, with one group labeled "Z." Because the pads at the IC sockets are also used as feed-throughs in



many cases, the (inexpensive) component integrated circuits can be soldered in directly on both sides of the board. The test sockets (as well as those for the other component ICs, if desired) may be wire-wrap sockets spaced off the board so that the pads may be soldered on both sides. The socket-to-board spacing can be adjusted so only the test sockets protrude from the eventual enclosure.

Control Line (D7)	Data Bit In	Socket Pin
0	D0	15
	D1	14
(loaded, but not	D2	13
latched)	D3	12
	D4	11
	D5	10
	D6	9
1	D0	7
	D1	6
(loaded, but not	D2	5
latched)	D3	4
	D4	3
	D5	2
	D6	1

Table 3. Data input to 16-pin test socket (from computer).

Control Line (D7)	Socket Pin	Data Bit Out
0	16 (V_{cc})	D0
	15	D1
	14	D2
	13	D3
	12	D4
	11	D5
	10	D6
	9	D7
	8 (gnd)	D0
1	7	D1
	6	D2
	5	D3
	4	D4
	3	D5
	2	D6
	1	D7

Table 4. Data output from 16-pin test socket (to computer).

tics for the board will be as shown in Table 5.

Table 5 may be used as a test for proper board operation with no test IC installed; the Board Test subroutine does just that.

The procedure for performing a particular test is as follows. There is an *executive* portion of the program that directs either the search tests or the explicit tests. This is generally followed by two-part library subroutines for each device described.

The first part is a device description. The second part contains the actual test. The latter also calls special subroutines in which the data bits are presented to and retrieved from the 8-bit I/O port.

It is in these data bit transfer routines that the user must make adjustments for patching to specific parallel I/O hardware (e.g., Mits 2PIO, 4PIO; Processor Technology 3P + S; etc.). These subroutines are well enough annotated in the listing that the user should have little difficulty determining where patch changes should be applied. The present arrangement is for a Mits 4PIO board with input/output located at ports 37 (in) and 39 (out). Note that subroutines are used extensively; common operations are shared between tests.

The structure of the program is conducive to easy addition of more device test subroutines as the need arises. The 7400 test subroutine may be used as a model. The price paid for this convenience is some loss in programming efficiency and speed. The speed reduction is really apparent in the search

```
2940 GOSUB 1750
2950 O(A)=X1
2960 GOSUB 4190
2970 IF O1(A)<>I(B) THEN GOTO 3030
2980 S9=1
2990 IF P9=1 THEN GOTO 3030
3000 PRINT B$
3010 PRINT O1(A)," (PIN ",A,") INVERTED EQUALS",NOT O1(A)," NOT",I(B)
3020 STOP
3030 NEXT X1
3040 RETURN
3050 REM ********************
3060 REM 7405 TEST ROUTINE
3070 GOSUB 1830
3080 PRINT "*** 7405 ***"
3090 PRINT "HEX INVERTERS WITH"
3100 PRINT "OPEN COLLECTOR OUTPUTS"
3110 GOTO 2830
3120 REM ********************
3130 REM 7406 TEST ROUTINE
3140 PRINT "*** 7406 ***"
3150 PRINT "HEX INVERTER BUFFERS/DRIVERS WITH"
3160 PRINT "OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS"
3170 GOTO 2820
3180 REM ********************
3190 PRINT "*** 7407 ***"
3200 REM 7407 TEST ROUTINE
3210 PRINT "HEX BUFFER/DRIVERS WITH"
3220 PRINT "OPEN-COLLECTOR HIGH VOLTAGE OUTPUTS"
3230 REM UNKNOWN ENTRY POINT
3240 A=1\B=2\GOSUB 3320
3250 A=3\B=4\GOSUB 3320
3260 A=5\B=6\GOSUB 3320
3270 A=9\B=8\GOSUB 3320
3280 A=11\B=10\GOSUB 3320
3290 A=13\B=12\GOSUB 3320
3300 IF P9=0 THEN PRINT C$\RETURN
3310 REM ********************
3320 REM BUFFER TEST (14 PIN)
3330 FOR X1=0 TO 1
3340 IF P9*S9=1 THEN GOTO 3440
3350 GOSUB 1750
3360 O(A)=X1
3370 GOSUB 4190
3380 IF O1(A)=I(B) THEN GOTO 3440
3390 S9=1
3400 IF P9=1 THEN GOTO 3440
3410 PRINT B$
3420 PRINT O1(A)," (PIN ",A,") BUFFERED EQUALS", O1(A)," NOT",I(B)
3430 STOP
3440 NEXT X1
3450 RETURN
3460 REM *******************
3470 REM 7408 TEST ROUTINE
3480 PRINT "*** 7408 ***"
3490 PRINT "QUADRUPLE 2-INPUT"
3500 PRINT "POSITIVE-AND GATES"
3510 REM UNKNOWN ENTRY POINT
3520 A=1\B=2\C=3\GOSUB 3580
3530 A=4\B=5\C=6\GOSUB 3580
3540 A=9\B=10\C=8\GOSUB 3580
3550 A=12\B=13\C=11\GOSUB 3580
3560 IF P9=0 THEN PRINT C$\RETURN
3570 REM ********************
3580 REM TWO INPUT AND GATE TEST (14 PIN)
3590 FOR X1=0 TO 1
3600 FOR Y1=0 TO 1
3610 IF P9*S9=1 THEN GOTO 3720
3620 GOSUB 1750
3630 O(A)=X1\O(B)=Y1
3640 GOSUB 4190
3650 IF O1(A)*O1(B)=I(C) THEN GOTO 3720
3660 S9=1
3670 IF P9=1 THEN GOTO 3720
3680 PRINT B$
3690 PRINT O1(A)," (PIN ",A,") OR ",O1(B)," (PIN ",B,") EQUALS ",
3700 PRINT (O1(A) AND O1(B))," , NOT ",I(C), "(PIN ",C,")"
3710 STOP
3720 NEXT Y1
3730 NEXT X1
3740 RETURN
3750 REM ******************
3760 REM 7409 TEST ROUTINE
3770 PRINT "*** 7409 ***"
3780 GOSUB 1830
3790 PRINT "QUADRUPLE 2-INPUT POSITIVE-AND"
```

(V_{CC} or gnd) = DI0	(control bit 0 or 1)
DO0→DI1	DO4→DI5
DO1→DI2	DO5→DI6
DO2→DI3	DO6→DI7
DO3→DI4	DO7 = (control)

Table 5. Input/output characteristics with no test IC installed.

mode. Conversion to machine language should not be difficult, and the increase in speed would be 10-100X.

Performing a Test

In this section we will briefly examine how tests are performed.

As a first example, a good IC (in this case, an SN7400 quadruple two-input NAND gate) was placed in the test socket and probed. See Run A. The test was successful . . . as expected. Pin 6, an output, was then bent away from the socket, and the chip was again examined. See Run B. A fault is indicated among the logic levels on pins 4, 5 and 6. When an error is found, the program automatically stops. In this case you might guess that pin 6 was open as the error occurred only when the output at pin 6 was supposed to go to a zero state.

If instead pin 6 is shorted to ground, the results shown on Run C are obtained. In this case the source of the fault is more obvious.

As an example of determining the device number of an "unknown" integrated circuit, an SN7404 hex inverter was placed in the test socket and a "search" performed. The program successfully discovered the identity of the mystery IC.

Port Expansion

The test circuit may also be readily adapted for use as a parallel input and output port expander by removing the 5 volt

```
3800 PRINT "GATES WITH OPEN-COLLECTOR OUTPUTS"
3810 GOTO 3520
3820 REM ******************
3830 REM 7410 TEST ROUTINE
3840 GOSUB 1830
3850 PRINT "*** 7410 ***"
3860 PRINT "TRIPLE 3-INPUT"
3870 PRINT "POSITIVE-NAND GATES"
3880 REM UNKNOWN ENTRY POINT
3890 A=1\B=2\C=13\D=12\GOSUB 3940
3900 A=3\B=4\C=5\D=6\GOSUB 3940
3910 A=9\B=10\C=11\D=8\GOSUB 3940
3920 IF P9=0 THEN PRINT C$\RETURN
3930 REM ********************
3940 REM THREE INPUT NAND GATE TEST (14 PIN)
3950 FOR X1=0 TO 1
3960 FOR Y1=0 TO 1
3970 FOR Z1=0 TO 1
3980 IF P9*S9=1 THEN GOTO 4090
3990 GOSUB 1750
4000 O(A)=X1\O(B)=Y1\O(C)=Z1
4010 GOSUB 4190
4020 IF O1(A)*O1(B)*O1(C)=NOT I(D) THEN GOTO 4090
4030 S9=1
4040 IF P9=1 THEN GOTO 4090
4050 PRINT B$
4060 PRINT O1(A)," (PIN",A,"), NAND",O1(B)," (PIN",B,"), NAND",
4070 PRINT O1(C)," (PIN",C,") EQUALS",NOT(O1(A)*O1(B)*O1(C))," NOT",I(D)," (PIN",D,")"
4080 STOP
4090 NEXT Z1
4100 NEXT Y1
4110 NEXT X1
4120 RETURN
4130 REM ********************
4140 PRINT "NO MATCH FOUND"
4150 PRINT "IN PRESENT LIST"
4160 STOP
4170 REM ******************
4180 REM UTILITY SUBROUTINE (14 PIN)
4190 GOSUB 910\GOSUB 1520\GOSUB 1040
4200 RETURN
READY
```

```
***INSTRUCTIONS***


TURN TESTER POWER OFF
READY?Y
INSERT IC TO BE TESTED
TURN POWER ON
READY?Y
ERROR
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
CHECK EITHER CABLE OR POWER SUPPLY
READY?
?Y
TURN TESTER POWER OFF
READY?Y
INSERT IC TO BE TESTED
TURN POWER ON
READY?Y


SEARCH OR TEST (S/T)?T
INPUT THE DEVICE NUMBER: ?8000
IC TESTER CHECKOUT ROUTINE
REMOVE IC FROM TEST SOCKET
READY?Y

TEST COMPLETED SUCCESSFULLY


TURN TESTER POWER OFF
READY?Y
INSERT IC TO BE TESTED
TURN POWER ON
READY?Y


SEARCH OR TEST (S/T)?T
INPUT THE DEVICE NUMBER: ?7400


*** 7400 ***


QUADRUPLE 2-INPUT
POSITIVE-NAND GATES
TEST COMPLETED SUCCESSFULLY
```

Run A. Sample run. Power to tester was missing. When applied, the program went on to examine the tester itself and then a good SN7400. Note: 8000 is the code for the "board test" subroutine.

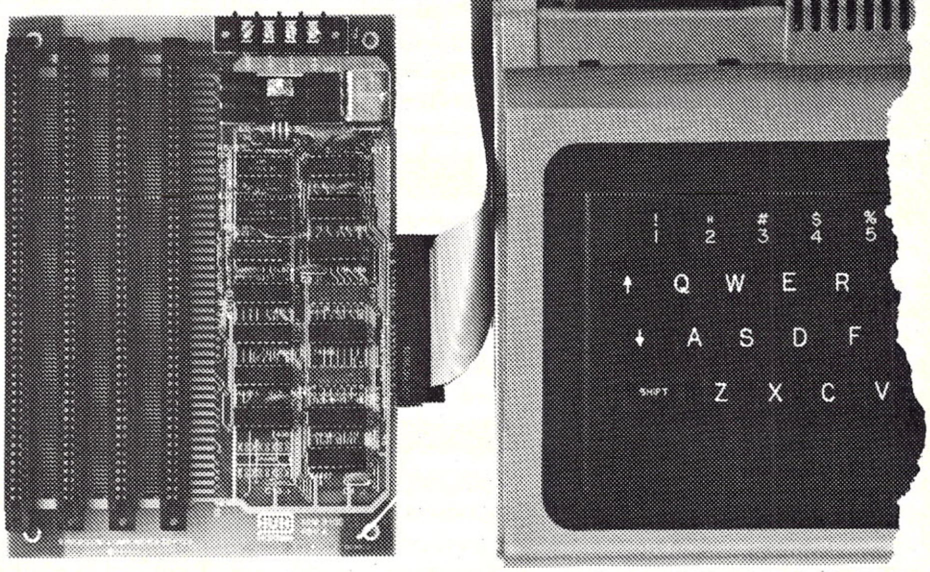
```
***INSTRUCTIONS***


TURN TESTER POWER OFF
READY?Y
INSERT IC TO BE TESTED
TURN POWER ON
READY?Y


SEARCH OR TEST (S/T)?T
INPUT THE DEVICE NUMBER: ?7400


*** 7400 ***


QUADRUPLE 2-INPUT
POSITIVE-NAND GATES
ERROR
 1 (PIN  4) NOR  1 (PIN  5) EQUALS  0 , NOT  1(PIN
STOP IN LINE 1400
READY
```

Run B. Integrated circuit tested for Run A is reexamined, but with pin 6 bent away from socket; an open output.

HUH ELECTRONICS ANNOUNCES THE

MINI-8100

AN S-100 BUS ADAPTER/MOTHERBOARD FOR THE TRS-80*



MINI-SIZE!

MINI-PRICE!

but MAXI-PERFORMANCE!

HUH Electronics' 8100 is the established leader in S-100/TRS-80 interfaces, running reliably since November 1978 and giving you the most features for your dollar. But some folks don't need all the 8100's versatility or want something more compact.

So we created the **MINI-8100.** The **MINI-8100** has the same proven reliable S-100 Bus interface circuitry as it's big brother coupled with a 4 slot shielded motherboard. It connects to the TRS-80 or to the expansion interface and opens up the vast world of S-100 Bus products to the TRS-80 owner.

For those of you who already own an S-100 mainframe or motherboard, the **MINI-8100** may be purchased without the built-in motherboard, (**MINI-8100S**). Just plug it in and go!

The **MINI-8100** has the same high quality as all HUH Electronics products—easy assembly, fully socketed, solder mask and silk screen component legend, comprehensive manual and design excellence that speaks for itself.

PRICES

MINI-8100 KIT	- includes all parts and one S-100 connector	- **$115**
MINI-8100 ASM	- assembled and tested, includes four S-100 connectors	- **$155**
MINI-8100S KIT	- S-100 sized board, plugs into mainframe	- **$ 95**
MINI-8100S ASM	- same as above, but assembled and tested	- **$125**

HUH ELECTRONICS

1429 Maple St.
San Mateo, CA
94402
(415) 573-7359

ORDERING INFO: The **MINI-8100** is available from leading computer dealers or factory direct. Please include $4 for shipping and handling. CA residents include 6% sales tax. We accept VISA or Mastercharge or we can ship COD.

USA DOMESTIC PRICES ONLY.

DEALER INQUIRIES INVITED

*TRS-80 is a Radio Shack® product.

Reader Service—see page 179

and ground leads to the test sockets. The first step in expansion is shown in Fig. 4. The inputs to be compressed and the expanded outputs are all connected through the 16-pin DIP socket, which was previously used for the IC test.

Note, however, that input and output compression/expansion can not be performed simultaneously through the single 16-pin socket. Thus, unless the final data receiver is strobed, data input to the socket from the external world will confuse the data transfer in the other direction. (The problem is only with the data output to the socket by the tester.)

The next step in expansion is shown in Fig. 5. In this case the output expansion is six bits to 24 bits and the input compression is 32 bits to 8 bits. Again the restrictions regarding simultaneous input and output apply.

The major value of this use is probably in the compression mode. For example, two 4-digit BCD (binary coded decimal) encoded panel meters (e.g., Analogic Model 2500) may be monitored through one 8-bit computer port using the second level of compression.

It may be argued that the Mits 4PIO board could instead be used to accomplish the same level of compression . . . and with greater speed since the software overhead would be less. Also, the total cost might be less. However, the Mits 4PIO board uses up 16 port addresses. This technique could work with a single PIO that used only two (one conventionally for a control channel) addresses.

Also, in terms of the compression of large input arrays, the number of Mits boards that can be crammed into an Altair 8800A, B or Imsai mainframe is perhaps six, depending on how many of the limited number of bus slots are filled with memory, SIO, video display, disk controller, etc., boards. (The Mits 680b is *very* limited in this respect.) If six such boards are allowed, the maximum number of four-digit panel meters that can be input is four per board, or 24 total. (Note that with the Mits 4PIO board, *all* I/O ports can be assigned to inputs. Thus, 64 bits may be input through each board.)

Most hobbyists will never get near this problem, but industrial control applications often have requirements that exceed this. The input port compression available with the multiplexing scheme is 256 bytes multiplexed into one byte; 128 four-digit panel meters can be monitored by one eight-bit parallel I/O port.

However, if monitoring such a large input array were really required, much more efficient multiplexing designs are available. In fact, a 6800-based microcomputer could serve as a multiplexer since it treats I/O as just another memory transfer.

In-Circuit IC Tester

Quite often the examination to be performed is not one of testing loose ICs as to what they are, but rather whether or

```
***INSTRUCTIONS***


TURN TESTER POWER OFF
READY?
INSERT IC TO BE TESTED
TURN POWER ON
READY?


SEARCH OR TEST (S/T)?S
*** 7404 ***
HEX INVERTERS
```

Run D. An SN7404 IC is placed in the test socket and a "search" is performed. The program successfully determines what the device number is.

```
***INSTRUCTIONS***


TURN TESTER POWER OFF
READY?Y
INSERT IC TO BE TESTED
TURN POWER ON
READY?Y


SEARCH OR TEST (S/T)?T
INPUT THE DEVICE NUMBER: ?7400


*** 7400 ***


QUADRUPLE 2-INPUT
POSITIVE-NAND GATES
ERROR
 0 (PIN  4) NAND  0 (PIN  5) EQUALS  1 , NOT  0(PIN  6)
STOP IN LINE 1470
READY
CONT


ERROR
 0 (PIN  4) NAND  1 (PIN  5) EQUALS  1 , NOT  0(PIN  6)
STOP IN LINE 1470
READY
CONT


ERROR
 1 (PIN  4) NAND  0 (PIN  5) EQUALS  1 , NOT  0(PIN  6)
STOP IN LINE 1470
READY
CONT


TEST COMPLETED SUCCESSFULLY
```

Run C. Same as Run A, but with pin 6 connected to ground; grounded output. Note: The program stops when an error is found. It can be "continued" to find more errors.

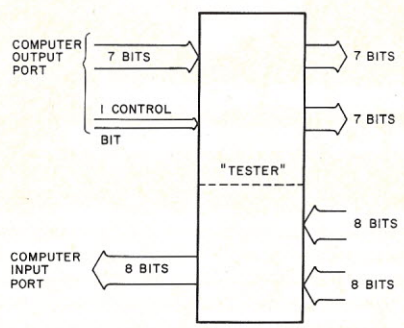


Fig. 4. First level of port expansion. Expansion out of the computer is 7 to 14 bits. Compression into the computer is 16 to 8 bits.

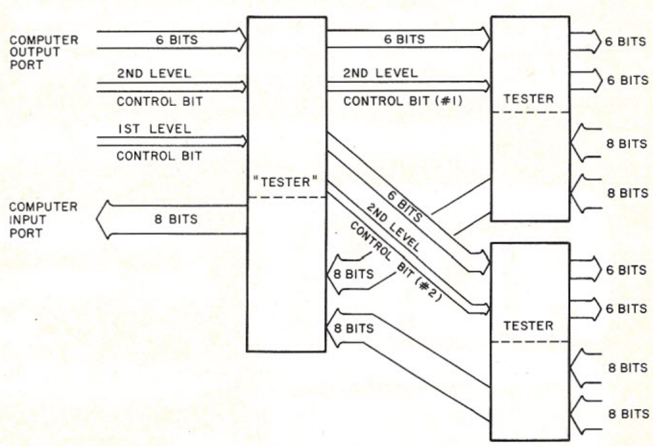


Fig. 5. Second level of port expansion. Expansion out of the computer is 6 to 24 bits. Compression into the computer is 32 to 8 bits.

not the IC of interest is correctly working in a circuit given the original load conditions. Also, the IC to be probed may be soldered in place.

In this situation, making the test without removing the component would be beneficial. (I haven't had a chance to implement this yet. However, the technique is easy to understand and is employed in at least one commercial test unit. Fluke.) In many cases this can be accomplished by using two IC tester boards and comparing the in-circuit component with one that is previously tested and found good. This may be done as follows.

A test cable (a ribbon cable with male DIP connectors on either end should be workable) is run between the first tester and the component to be tested. The IC to serve as a reference is placed in the corresponding socket of the second tester.

Using a simple (preferably in machine language) program (not presented here), you can monitor the inputs to the in-circuit component and place these same logic levels on the inputs of the reference device. The outputs are then compared and discrepancies noted.

A few caveats are in order, however. The first is that the logic level changes on the in-circuit IC must be slow enough so that the software tests may keep up with the changes.

Second, Tri-state (and low-current *source* outputs) devices may not be correctly tested in some cases.

Third, and very important, a discrepancy may not mean that the IC being tested is faulty, but rather that the load on its outputs may exceed its capability (e.g., shorted input on the next component in the logic line). However, this technique could at least be a fast way to troubleshoot in a manner similar to using a much more expensive logic probe.

Conclusions

The circuitry discussed above has several characteristics that represent a compromise between cost, complexity, reliability and versatility. If any of these attributes were particularly emphasized, the design would have been different. For example, the circuit layout complexity could be significantly reduced using simple IC 8-bit latches such as the 74100 or 8212. However, cost and reliability (the 8212 is an MOS device) would be affected.

I believe that sufficient information has been given in this article to enable most small users to construct and operate an IC tester for use with their own microcomputers—given that they have an 8-bit I/O port.■

Whip File Wipeouts

This short article shows how to append another file with CLOAD in the Level II TRS-80.

Roger L. Pape
7545 Marble Dr.
Liverpool NY 13088

There are two 16-bit pointers in the TRS-80 Level II table space that control the storage of BASIC source statements. The value starting at $40A4_H$ (16548_{10}) points to the start of the BASIC statement storage area; whereas, the value starting at $40F9_H$ (16633_{10}) points to two bytes past the last statement, i.e., where the next line number would be stored. (Note that the standard Z-80 convention of storing the lower byte first is used.)

The first pointer is initialized at power-up to $42E9_H$ (17129_{10}) and normally remains at this value. It is used by the CLOAD routine as the point to start storing the BASIC file read from tape. Normally, a second CLOAD will pick up the same value and store the new file over the data already in memory, wiping out the first file. Overcoming this problem is simply a matter of changing the lower pointer before the second CLOAD.

Modification Steps

First, print out the two bytes of the end pointer.

```
PRINT PEEK(16633),PEEK(16634)
```

Subtract 2 from the first value (i.e., the lower byte). If this results in a negative value, borrow from the high byte (i.e., subtract 1 from the second value and add 256 to the first). These bytes are then poked into the start pointer locations:

```
POKE 16548,lb:POKE 16549,hb
```

where lb and hb are the adjusted low and high bytes, respectively.

You can then CLOAD a second file beyond the source statements that are already in memory. After this file is read in, the starting pointer is restored to its original value:

```
POKE 15648,223:POKE 16549,66
```

The result is a concatenated file with both sets of statements. A sample case showing the required steps is shown in Example 1.

```
>PRINT PEEK(16633),PEEK(16634)
206          74
>POKE 16548,204:POKE 16549,74
READY
>CLOAD
READY
>POKE 16548,233:POKE 16549,66
READY
>
```

Example 1.

If you examine the format used to store BASIC source code in memory, you will see that the first two bytes of each line comprise a link pointer to the start of the next line (i.e., its absolute memory location). Obviously, when a file is stored at a new starting location, these pointers must all be modified. Fortunately, the system software will automatically do this if the above procedure is followed. At the end of a CLOAD, the system scans the data read in and updates all these link pointers before reading the next command.

One precaution must be observed when you use this procedure: The line numbers for the second file loaded should all be greater than those already in memory. In other words, enter the source data in ascending line-number order!■

HUH?

The Model 8100 TRS-80 to S-100 adapter/motherboard from HUH Electronics will help eliminate some of the communications problems that still exist in the microcomputer world.

Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

The lack of standards in personal computing has always distressed me. There are so many buses, cassette-tape formats, disk formats and BASIC variations that it seems each of us is speaking a different language. And in a sense, I guess we are.

For my own personal standards I've chosen the S-100 bus, the CP/M Disk Operating System and Microsoft Disk BASIC. I settled on these because they seem to be the most popular of the large number of choices available. Most popular does not necessarily mean best, but it does mean that I will have access to the largest possible amount of easily usable hardware and software. However, I also have a TRS-80 that has to fit into my scheme of things somewhere.

HUH Electronics (1429 Maple St., San Mateo CA 94402) has recently come out with their Model 8100, which is a TRS-80 to S-100 adapter/motherboard that can give you the best of two worlds. It's like being fluent in two different languages. You can use all of your TRS-80 software and hardware and also have access to most (but not all, as we shall see!) of the hardware and software designed for the S-100 bus.

The 8100, depicted in the accompanying photo, can be purchased in many configurations, both in kit form and completely assembled and tested as shown in Table 1. The basic board comes with the S-100 interface circuitry and one S-100 connector. Also available are a RAM option and an I/O option. It may be hard to tell from the photo, but this is a large board; it measures 14 × 17 inches.

Item	Kit Price	Assem. Price
Basic unit	$185	$245
RAM Support Option	45	75
I/O option	85	115
5 connectors and guides	45	75
Total if purchased separately	$360	$510
Total if purchased together	$295	$375

Table 1. The 8100 price list and options. Note that the basic unit contains the S-100 interface and one S-100 socket and card guide. The RAM support option does not include the memory chips. These are available from various sources for well under $100 for 16K.

Features

Because the features that you get depend on the options that you order, we'll discuss features as they pertain to the various sections of the board.

The basic board. This contains the S-100 bus interface circuitry that takes the TRS-80 expansion bus and buffers, modifies or generates the various signals that are needed by the S-100 bus. In other words, this is all you need to start using S-100 boards with your TRS-80. I've tried quite a few different S-100 boards, which we'll discuss in the Compatibility section.

The RAM support option. This allows you to use 4K or 16K ICs for memory expansion. For instance, what did you do with those 4K chips you removed while upgrading your keyboard unit to 16K? They can go in here, or you can buy 16K ICs. If you have an S-100 RAM board or two lying around, then you don't need this option.

The I/O option. A parallel and a serial port are provided. The parallel port is bidirectional and includes input and output strobes for handshaking. These strobes can be set up for either active-low or active-high operation, depending upon the needs of your peripherals. Power (+5 and +12) is available at the parallel port connector, but I wouldn't advise loading it down too much.

The serial port can be configured for either RS-232 or 20 mA current loop operation, and the TRS-80 can either be a computer or a terminal. This means that it is a computer when connected to its peripherals and a terminal when connected to a larger computer. This port uses a National INS-8250 UART, which is completely software programmable. Any baud rate between control features are between 50 and 56,000 can be selected. Various interrupt and modem control features are also available. An 8-page INS-8250 UART manual is part of the *8100 User Manual.*

The I/O section uses eight switch-selectable port addresses to handle various control, sense and input-output functions. Software examples of how to read and write to both the serial and parallel ports are spelled out. Software to drive a serial RS-232 printer from the BASIC LLIST and LPRINT statements is also provided.

The 8100 has edge-connector expansion ports for both the TRS-80 bus and the S-100 bus. This means that you can plug any accessories onto the 8100 that are now plugged into the keyboard unit and that you can add an S-100 motherboard, such as the Godbout, Artec or

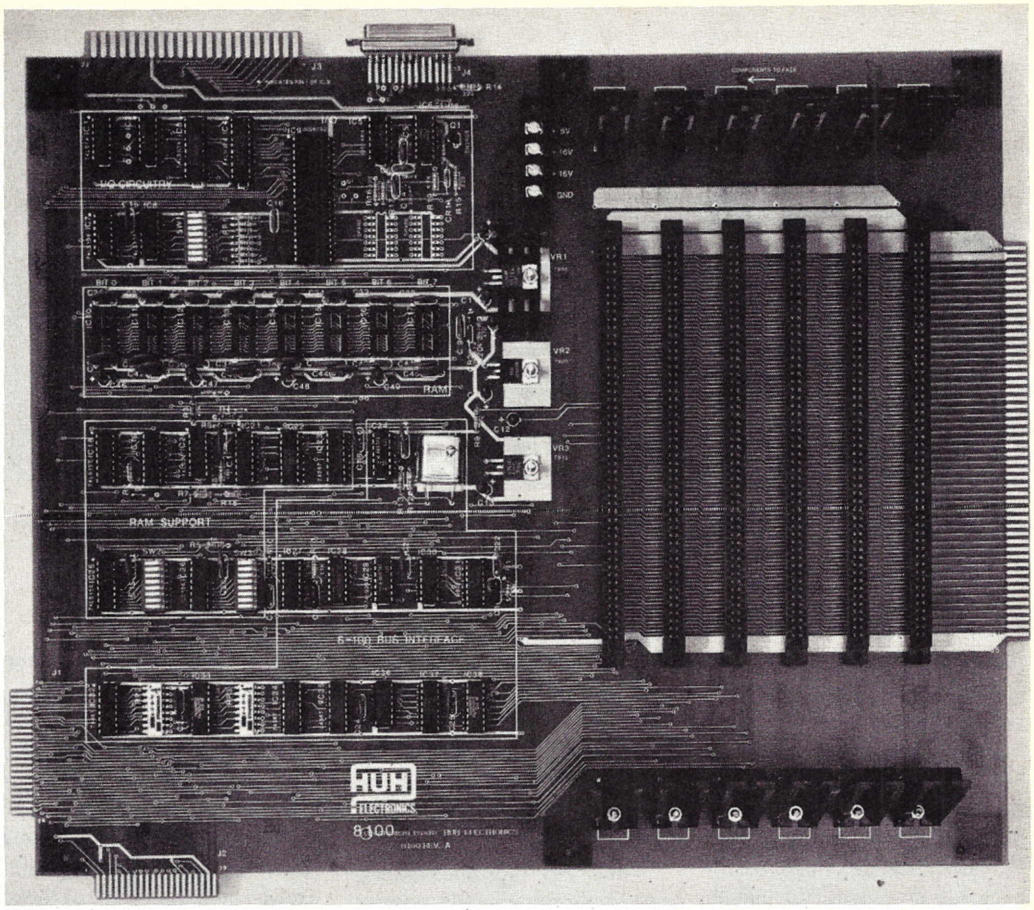
Vector, for even more S-100 slots.

Assembly

My 8100 arrived assembled, but I did take some time to study the assembly instructions to get some idea of what the kit-builder would have to contend with. I have built literally dozens of kits of all sizes, including two 25-inch color TV sets, and I feel well qualified to pass judgement on these instructions.

HUH has done a good job, and I think that even a beginning kit-builder can handle the 8100. There are no hidden tricks in its construction, and the step-by-step instructions don't leave anything to chance. An errata sheet details two hardware changes and two manual changes that the user must make.

First comes a how-to section: how to install capacitors, resistors, diodes and IC sockets. By the way, those sockets are a definite plus. It isn't much fun trying to unsolder a defective IC; but even more important than that, the easiest way to find that defective IC in the first place is by substitution. Try figuring out which IC is bad by substitution when you have 30, 40 or 50 of them all soldered in.



The HUH 8100 TRS-80 to S-100 adapter board. Note that the board is marked off into sections according to function. The S-100 expansion connector is to the right of the six S-100 sockets, and the TRS-80 expansion connector is on the lower left-hand edge. Power enters at the terminal strip located at the top center, and the parallel and serial I/O connectors are on the top, to the left. The cable from the TRS-80 attaches on the bottom left.

The board is marked off into sections, and you build the sections for the options that you've purchased. These are the RAM support, I/O and S-100 bus interface sections discussed above, which make this more than just an S-100 adapter board.

If you experience difficulty after assembling your kit, you can turn to the dealer or the factory for assistance. Kits are warranted for parts, while factory-assembled units are warranted for both parts and labor. Incidentally, Mark Garetz, the president of HUH Electronics, assures me that every factory-assembled 8100 is completely tested for all functions and that boards that cannot pass all tests do not leave the factory. He claims to have a low defective-board-return rate as a result of this policy.

Or, if you have or want the experience, you can use the extensive and clearly written Theory of Operation chapter and the large schematics to find your own problems. I did just that after an undersized fuse in my −16 volt power supply failed.

Installation

The 8100 is a complete unit that needs only a power supply to begin operation. Power requirements are +8 volts at 1 Amp, +16 volts at 500 mA and −16 volts at 500 mA. These satisfy the needs of the 8100; to this must be added the requirements of any boards that are plugged into the S-100 slots. You can buy commercial power supplies to provide these voltages, and one is recommended in the manual. You can also build your own if you're handy that way or buy one on the surplus market. Mine is home brew, left over from a long-forgotten project.

After your power supply has been tested and connected, all you have left is the connection to the TRS-80 keyboard expansion bus. This connection is made with a 40-conductor ribbon cable (with all power turned off, of course!). I would imagine that you could also connect it to the expansion interface, but since the 8100 duplicates or improves on most of the features of the expansion interface, I can't see any reason to have both.

With everything powered up, the first test is for normal operation of the TRS-80. If that is OK then you can start trying out various S-100 boards for compatibility. The manual gives some clues on this subject.

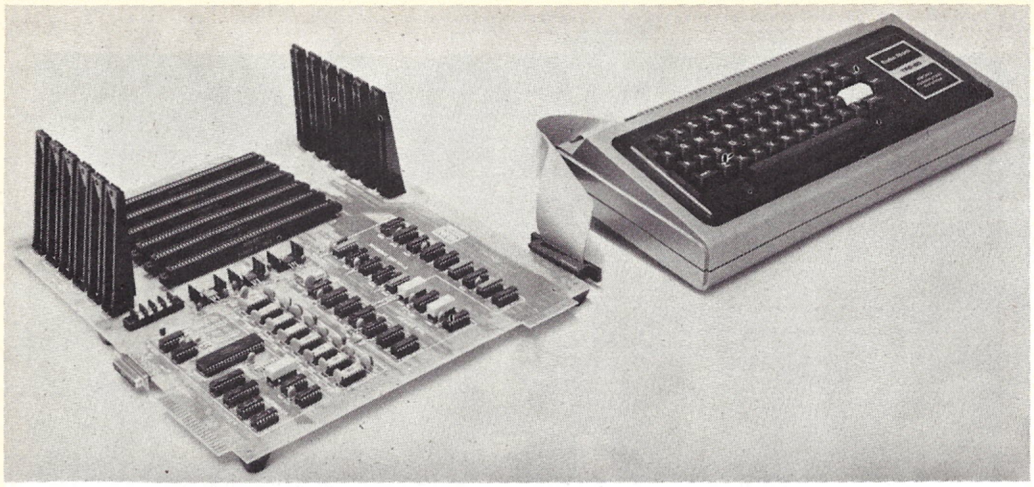
Compatibility

There are two strict rules for compatibility that will eliminate some S-100 boards: no DMA (direct memory access) boards and no boards that would halt the TRS-80 processor for more than 1 millisecond. Both of these rules are a result of the refresh requirements of the dynamic RAM memory ICs. If you stop the processor too long, they'll forget everything they know!

I first tried three different 16K static RAM boards addressing each one to start at 32768, which is the normal top of the 16K TRS-80. In effect, I now have a 32K machine. PRINT MEM answers 31956 instead of 15572. Quite an improvement!

Using my assembly-language operating system enables me to enter or alter or display or execute assembly-language programs in either the TRS-80 or the 8100 memory space.

Another board that I slipped into a slot is the MicroSounder by American Micro Products (6550 Tarnef M/S 11, Houston



The HUH Electronics 8100 TRS-80 to S-100 adapter/motherboard connected to the TRS-80 keyboard-CPU unit.

TX 77074). This is a unique and outstanding example of what we can expect as integrated circuits become more and more sophisticated and complex. Without dwelling too long on the features of the Micro-Sounder, which is built around the Texas Instruments SN76477 complex sound IC, I'd like to briefly describe it for you.

Various sound parameters can be software controlled so, theoretically, it should be able to simulate any and all sounds. Within a very few minutes after it arrived, I had it making siren, bird, train, machine gun and phasor noises. In any case, it works just as well in the 8100 as it does when it's plugged into my Z-2.

The PerCom CI-812 Cassette-I/O board also works quite well. This board contains both a serial RS-232 port and a Kansas City cassette interface. By a skillful manipulation of software it should be possible to read just about any tape format into or out of your TRS-80.

The Software Technology "Music System" would not work at all since it relies on the INTE lead (S-100 pin 28), which is not implemented on the 8100. However, it should be possible to use some other lead or even one bit from the parallel port as I did in my SOL.

Another board that gave me trouble is the Cromemco D+7 I/O interface. This is made up of a bidirectional parallel port, seven digital-to-analog channels and seven analog-to-digital channels. The D+7 is looking for a 2 MHz clock on pin 24 but instead finds 1.76 MHz from the TRS-80. A 2 MHz clock is available on pin 49, which can be strapped to the clock input in place of pin 24.

Since the digital-to-analog and analog-to-digital conversions take time, the processor is put into a wait state (11 cycles, 5.5 microseconds) while the conversions take place.

The processor normally holds the PHLDA lead (which is S-100 pin 26) low.

Without this low, the D+7 address decoder is disabled and the board does not respond. PHLDA is not implemented on the 8100, and pin 26 should be tied to ground.

Just out of curiosity, I tried my new Imsai VIO-C Video Interface board, but I could tell right away that it wasn't going to fly. It wants both phase 1 and phase 2 clocks at 2 MHz but won't find phase 1 anywhere on the bus. But then the TRS-80 has its own video interface and it doesn't really need another one.

And finally, I have a parallel I/O board which I built up on a Vector protoboard that drives my Malibu 160 line printer. Some superficial testing indicates that this board is OK, too. I can't use the TRS-80 to drive the line printer yet because I haven't yet rewritten the 2K printer driver software to fit the TRS-80.

That is the extent of my S-100 boards, except for the Discus Disk Controller from Thinker Toys. HUH claims that both the Discus and North Star Disk systems will work with the TRS-80 and the 8100. However, a little hardware and software modification is needed. Both of these DOS's need RAM in low memory, which in the TRS-80 is the position of the BASIC ROM. The hardware mod consists of a switch to disable the ROM while using the DOS. It is also necessary to relocate the keyboard scan and video driver routines to high memory. Software patches are given for both of these moves.

Don't let this bother you, though; there is a version of CP/M on the way that will be compatible with the TRS-80, which should solve that problem. I've never been sold on mini-disks, and the thought that I'll soon be able to plug my full-size floppies into the TRS-80 gives me quite a boost.

The manual states that most static RAM, EPROM and I/O boards should work and that the ultimate test is to see which ones will work on a trial basis. It especially suggests that you try out that new S-100 board you're considering before buying, if at all possible.

Conclusion

OK, I obviously feel that the HUH 8100 is a worthwhile accessory for the TRS-80 and that it is worth the asking price. Didn't I find anything that could be improved upon? Yes, a few mechanical points did come to my attention as I studied and used the 8100.

When it arrived by UPS, the box was crushed and two disk ceramic capacitors were broken. The packing material was crumpled-up newspaper, which is not sturdy enough to support the large box. I discussed this with HUH, who stated that they were going to start using those poly-whatever peanuts, which are much better and should solve that problem.

As I've mentioned, the 8100 is constructed on a large printed circuit board. As it stands, on its six rubber feet, it is completely exposed to the elements, kids, wire scraps and other potential hazards. The manual mentions that another manufacturer will soon be coming out with an enclosure. I would definitely want to buy or build something to protect my investment.

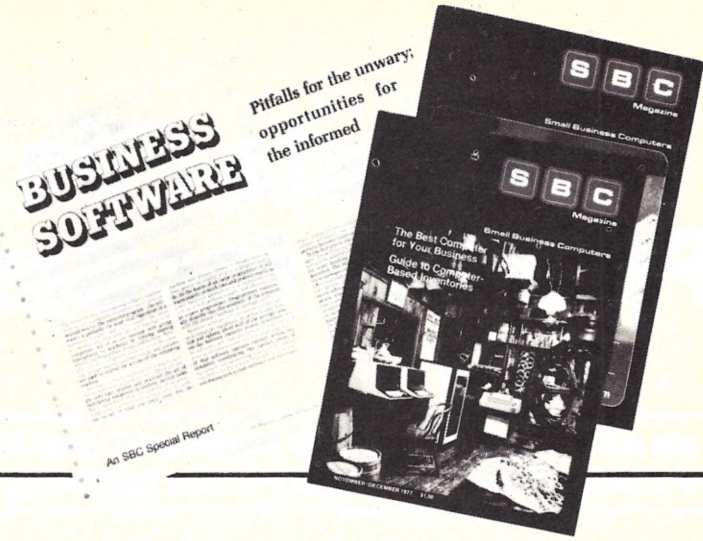
Also, there is not much support under the S-100 slots, and the 8100 flexes badly when boards are plugged in. I took care of this by moving two of the rubber feet to a position close to the S-100 sockets but would like to have had at least two more of these feet.

And finally, the card guides are bolted to the PC board with thin screws that could pull through the plastic if tightened too much or if you accidentally hit one of the guides. I substituted washers and larger, but not longer, screws. When mounted, the guides stand individually and are really in need of a little more support. A strip of plastic glued or bolted across the backsides of all six guides would make for a sturdy structure.

All of these comments relate to mechanical matters and, as far as I'm concerned, they don't detract from the value of the 8100. For the TRS-80 owner who wants to expand his machine and who wants the greatest variety of software, accessories and peripherals available to him, the HUH 8100 is a best buy. ■

Announcing . . .
Small Business Computers
Magazine

The magazine for
users and potential users
of small business computer
products and services



***Small Business Computers** Magazine (SBC)* is the monthly magazine for businessmen who are in the process of purchasing or installing their first computers. It is the bridge between the world of business and the world of small computers.

***Small Business Computers** Magazine* is a practical *how-to* publication . . . written in non-technical language . . . and stressing business applications for small computer systems. Each monthly issue includes:

FEATURE SURVEY REPORTS on particular areas of current interest, such as *Software Packages for Small Business Applications, Small Manufacturing Systems, Inventory Control Systems, Microcomputer Business Applications,* and so on.

APPLICATION STORIES: Real-life examples of computer applications in the small business environment — articles which stress computer capabilities and benefits, and what to watch for when purchasing and installing a computer.

COMPUTER PROFILES: Spotlight reviews of new computer systems for the small businessman.

IDEAS AND INNOVATIONS: Helpful items which enable more effective utilization of small business systems.

INFORMATIVE ADVERTISING: The leading mini and microcomputer companies are regular advertisers, and Small Business Computers Magazine is a showcase of small systems for the businessman.

SPECIAL CHARTER SUBSCRIPTION

$9 for 12 monthly issues

*Receive the next 12 issues of **Small Business Computers** Magazine at 50% off the cover price by entering your charter subscription today. Send your check along with the coupon to:*

SMALL BUSINESS COMPUTERS Magazine

33 Watchung Plaza • Montclair, NJ 07042

SMALL BUSINESS COMPUTERS Magazine ✓S69 KB-77
33 Watchung Plaza • Montclair, NJ 07042

☐ YES. **Enter my charter subscription at the ½-price cost of $9 for 12 monthly issues.**
☐ Check enclosed. ☐ Bill me.

Name ____________________

Organization ____________________

Address ____________________

City/State ____________________ Zip ____________

✓ Reader Service—see page 179

A Data-File Creation Program for Small Business

As we all know by now from experience and from reading, there's a lot of small-business software out there. Here's the beginning of a relatively inexpensive alternative.

Forest E. Myers
5114 Garnett St.
Shawnee KS 66203

I was sitting in the big, overstuffed chair with a soft drink in hand. In front of me was a big, wooden desk with two telephones (which I later found out rang almost continuously) and the usual clutter of paper. Behind the desk sat Bill.

Bill was saying, "I bought that thing (the microcomputer in the corner) to keep track of my property business. I tried a service bureau, but all they gave me were reports I didn't need, and those that I did need contained all the wrong numbers. That thing ought to be able to do the job; however, all that 'gett'n' and 'put'n' has me baffled. I need someone to show me how to use the BASIC interpreter to read and write data files. Once I get the file-creation business down, I can take it from there."

Bill's problem is not unknown to anyone who has had access to a computer. Given a computer-oriented problem, how do you get the computer to solve it? For Bill, the problem was development of programs to keep track of property revenues and expenses. The programs developed had to run on his 34K system (34K bytes of RAM, dual disks, CRT and hard-copy output device). They also had to be simple to use and capable of accepting a wide variety of financial information.

The Program

The solution to Bill's programming problem makes up the heart of this article. Although originally intended for property file creation, it can be readily modified for multi-department business applications. In fact, that is what has been done; the designation property number has been replaced by the label department number.

The file-creation program,

Program listing.

```
10  DIM A$(256),B$(10),C$(10)
20  REM BASIC REQUIRES THAT ALL STRINGS BE INITIALIZED BEFORE FIRST USE
30  A$=" " : FOR I=1 TO 8 : A$=A$+A$ : NEXT I
40  FOR I=1 TO 10 : B$=B$+" " : C$=C$+" " : D$=D$+" " : NEXT I
50  #""
60  FOR I=1 TO 7 : #" " : NEXT I
70  #"                    FILE CREATION PROGRAM"
80  #"                        VERSION 1.0"
90  #"                         10/22/78"
100  REM *******  A$ IS INPUT/OUTPUT STRING  *******
110  REM          B$ HOLDS FILE NAMES TO BE INPUT OR OUTPUT
120  REM          C$ USED TO HOLD ANSWER TO PROGRAM QUERIES
130  REM          D$ HOLDS SELECTION INFORMATION USED IN PRINT
140  REM           O HOLDS OPTION SELECTED
150  REM           D HOLDS INPUT/OUTPUT DEVICE SELECTED
160  REM           R HOLDS MAXIMUM NUMBER OF RECORDS IN FILE
170  REM           B HOLDS NUMBER OF INPUT/OUPUT BLOCKS NEEDED
180  REM          S1 SWITCH -- 1 IF EDIT OPTIONS ARE USED
190  REM          S2 SWITCH -- 1 IF CREATE OR ADD TO FILE OPTION USED
200  REM           K HOLDS NUMBER OF RECORDS CURRENTLY IN THE FILE
210  REM          R9 HOLDS USER INPUT FOR RECORD TO BE CORRECTED
220  REM           L LOOP COUNTER FOR BLOCK IN FILE CREATE OPTION
230  REM FILE RECORD ZERO HOLDS NUMBER OF RECORDS CURRENTLY
240  REM IN THE FILE, MAX NUMBER OF RECORDS THE FILE CAN HOLD,
250  REM AND THE INTERNAL FILE LABEL GIVEN TO THE FILE.
260  REM     RECORDS 1 THRU END HOLD ACTUAL DATA
270  REM RECORD FORMAT IS AS FOLLOWS
280  REM          POS 1-5  CHECK NUMBER
290  REM              6-7  DEPARTMENT NUMBER
300  REM              8-10 DISTRIBUTION NUMBER
310  REM              11-13 AMOUNT
320  REM RECORDS ARE BLOCKED INTO 14 RECORDS PER BLOCK
330  REM THAT MEANS THAT FOR EACH READ OR WRITE TO AN INPUT OR
340  REM OUTPUT DEVICE 14 RECORDS ARE READ OR WRITTEN.
350  FOR I=1 TO 5 : #" " : NEXT I
360  INPUT "Options   1. Create new file  2. Edit old file ",O
370  IF O<1 OR O>2 THEN #"Not valid option number"
380  IF O<1 OR O>2 THEN 360
390  IF O=2 THEN 570
400  INPUT "Enter input/output device number ( 0 or 1) ",D
410  INPUT "Enter name of file to be created ( 6 characters ) ",B$
420  IF LEN(B$)<>6 THEN #"File name not 6 characters in length "
430  IF LEN(B$)<>6 THEN 410
440  S2=1
450  B$=B$+".DA"
460  B$=B$+CHR$(0)
470  INPUT "Enter maximum records to be included in file ",R
480  B=INT(R/14)+1 : REM *** DETERMINING OUTPUT BLOCKS NEEDED
490  OPEN (0,E,B$,1,D,B)
500  #" " : #" " : #"MAKING SPACE ON DISK"
510  FOR I=0 TO B : PUT (0,E,A$,I) : NEXT I
520  #"DISK SPACE ALLOCATED"
530  K=0
540  L=0
550  GOSUB 1150 : REM GO TO DATA ENTRY ROUTINE
560  END
570  INPUT "Enter input/output device number (0 or 1) ",D
580  INPUT "Enter name of file to be edited ( 6 chrs ) ",B$
590  IF LEN(B$)<>6 THEN #"File name not 6 characters long"
600  IF LEN(B$)<>6 THEN 580
610  B$=B$+".DA"
620  B$=B$+CHR$(0)
630  OPEN (0,E,B$,3,D)
640  GET (0,E,A$,0)
650  C$=A$(1,3)
660  CONVERT C$ TO K
670  #" " : #" " : #" "
680  #"Number of records in file ",K
690  C$=A$(4,6)
700  CONVERT C$ TO R
710  #"Maximum records that can be in file ",R
720  #"Internal file label ",A$(7,44)
730  FOR I=1 TO 4 : #" " : NEXT I
740  INPUT "Options  1.Add to file  2.Edit record  3.List file  4.End ",O
750  IF O<1 OR O>4 THEN #"Not valid option number "
760  IF O<1 OR O>4 THEN 740
770  K1=K+1
780  B=INT(K1/14)
790  A=K1-B*14
800  IF A>0 THEN B=B+1
810  IF A=0 THEN A=14
820  IF O=2 THEN 930
830  IF O=3 THEN 1890
```

100	Assets	400	Revenue
	110 Cash		410 Sales
	120 Accounts Receivable		420 Other
	130 Furniture and Fixtures		
		500	Cost of Goods Sold
200	Liabilities		
	210 Mortgage Payable	600	Operating Expenses
	220 Accounts Payable		610 Interest Expense
			620 Utilities
300	Capital		630 Depreciation

Example 1. Sample chart of accounts.

Create, like so many other programs, is based upon checkbook entries, although other documents can be used. Create allows development of new data files and editing of files. It allows additions to old files, correction of individual records in old files and listing of files. Files can be listed in their entirety, or selected records can be listed.

Since the program is well documented, I will say little about its operation. However, some remarks are in order regarding the file-creation option. Regarding this option, Create initializes space on the storage device before any information is entered by the user. This initialization process requires the user to enter what he feels will be the maximum number of records placed in the file (a record is defined as a check number, department number, distribution number and amount).

By creating a blank file at the outset, the user can enter data as time allows, terminate entry and come back later to enter additional data to the file. Entering a "/" for a check number will end the current data-entry session. The user can pick up where he left off by using the add-to-file option in the edit mode. The large, blank file, therefore, allows you to break up data entry into several short sessions, rather than one long one, without having to create many small files. (Since tape and disk space are relatively inexpensive, leave yourself plenty of file room.)

A record is composed of a check number, a department number, a distribution number (revenue or expense code) and an amount. Five digits are reserved for the check number, two for the department number, three for the distribution number and eight for the amount. A record is, therefore, 18 bytes long. With the exception of the distribution number, most of the items are self-explanatory. The distribution number is the account code taken from the business's chart of accounts. (See Example 1.)

This briefly summarizes some of the features of the program. To further understand the program, a few words on the interpreter and its operation are in order.

The Interpreter

The interpreter used is "Business Basic" by Micro Works of Cincinnati OH. There is a digital cassette and disk version of this interpreter. It has many features, such as cursor positioning, on error, image and print using, which have not been used in the program listing. Instead, I only used those commands most widely available in other BASICs.

In Business Basic, input and output to storage devices is via strings (that's the reason for inputting everything in strings; it could be input as numeric information, but would have to be converted into string information to be written to the storage device). Additionally, the maximum number of bytes that are read or written at one time is 256 (one page). With the DOS on Bill's machine, one page is written or read whether one byte or 256 are used.

Since the record length for Bill's business application was 18 bytes (5-digit check number, 2-digit department number, 3-digit distribution number, 8-digit amount), 14 records were grouped together to form data

```
840  IF O=4 THEN 2390
850  REM ADD TO FILE OPTION
860  L=B
870  S1=1 : S2=1
880  GET (0,E,A$,B)
890  GOSUB 1230 : REM GO TO DATA ENTRY ROUTINE
900  GOSUB 1710 : REM UPDATE RECORD ZERO FOR ADDITIONAL ENTRIES
910  GOTO 730
920  REM ********************  EDIT RECORD ROUTINE  ***************
930  INPUT "Enter record to be corrected ",R9
940  S1=1 : S2=0
950  IF R9>R THEN #"Only a maximum of ";R;" records are in this file"
960  IF R9>R THEN 930
970  IF R9>K THEN #"Last record currently used in the file is";K
980  IF R9>K THEN 930
990  B=INT(R9/14)
1000  A=R9-B*14
1010  IF A>0 THEN B=B+1
1020  IF A=0 THEN A=14
1030  GET (0,E,A$,B)
1040  M=A*18
1050  #"Record ";R9
1060  #"Check No. ";A$(M-17,M-13)
1070  #"Dept No.  ";A$(M-12,M-11)
1080  #"Dist No.  ";A$(M-10,M-8)
1090  #"Amount    ";A$(M-7,M)
1100  GOSUB 1460 : REM GO TO ERROR CORRECTION ROUTINE
1110  INPUT "Another to be corrected ? ",C$
1120  IF C$="Y" THEN 930
1130  GOSUB 1710 : GOTO 730
1140  REM  *********************  DATA ENTRY ROUTINE  *************
1150  REM           L  HOLDS BLOCK NUMBER TO BE WRITTEN OUT
1160  REM           R  HOLDS MAX NUMBER OF RECORDS IN FILE
1170  REM           K  HOLDS CURRENT NUMBER OF RECORDS IN FILE
1180  #" " : #" " : #" "
1190  #"No decimals are to be entered as part of AMOUNT"
1200  #" " : #" " : #" "
1210  L=L+1
1220  A=1
1230  REM ENTER POINT FROM ADD TO FILE OPTION
1240  FOR J=A TO 14
1250    K=K+1
1260    #"RECORD ";K
1270    IF K>R-4 THEN #"Room for only";R-K;" more records"
1280    M=J*18
1290    INPUT "Enter CHECK NO. (5 digits max) ",A$(M-17,M-13)
1300    IF A$(M-17,M-17)="/" THEN EXIT 1400
1310    INPUT "Enter DEPT NO. (2 digits max) ",A$(M-12,M-11)
1320    INPUT "Enter DIST NO. (3 digits max) ",A$(M-10,M-8)
1330    INPUT "Enter AMOUNT   (8 digits max) ",A$(M-7,M)
1340    INPUT "Corrections (Y or N ) ",C$
1350    IF C$="Y" THEN GOSUB 1460 : REM GO TO ERROR ROUTINE
1360    NEXT J
1370  PUT (0,E,A$,L)
1380  A$=" " : FOR I=1 TO 8 : A$=A$+A$ : NEXT I
1390  GOTO 1210
1400  A$(M-17,M-13)="     "
1410  K=K-1
1420  PUT (0,E,A$,L)
1430  GOSUB 1710
1440  RETURN
1450  REM ****************   ERROR CORRECTION ROUTINE   ************
1460  INPUT "Enter first two letters of item to be corrected ",C$
1470  IF C$(1,2)="CH" THEN 1540
1480  IF C$(1,2)="DE" THEN 1570
1490  IF C$(1,2)="DI" THEN 1600
1500  IF C$(1,2)="AM" THEN 1630
1510  IF C$(1,2)="ES" THEN 1650
1520  #"Not valid item name"
1530  GOTO 1460
1540  A$(M-17,M-13)="     "
1550  INPUT "Enter new CHECK NO. ",A$(M-17,M-13)
1560  GOTO 1660
1570  A$(M-12,M-11)="  "
1580  INPUT "Enter new DEPT NO. ",A$(M-12,M-11)
1590  GOTO 1660
1600  A$(M-10,M-8)="   "
1610  INPUT "Enter new DIST NO. ",A$(M-10,M-8)
1620  GOTO 1660
1630  A$(M-7,M)="        "
1640  INPUT "Enter new AMOUNT ",A$(M-7,M)
1650  #" "
1660  INPUT "More corrections to this record ? ",C$
1670  IF C$="Y" THEN 1460
1680  IF S2=1 THEN 1700
1690  PUT (0,E,A$,B)
1700  RETURN
1710  REM UPDATE RECORD ZERO
1720  REM RECORD LAYOUT FOR RECORD ZERO
1730  REM         POS  1-3    NO. OF RECORDS CURRENTLY IN FILE
1740  REM         POS  4-6    MAX NUMBER OF RECORDS THAT CAN BE IN FILE
1750  REM         POS 7-44    INTERNAL FILE LABEL
1760  A$=" " : FOR I=1 TO 8 : A$=A$+A$ : NEXT I
1770  IF S1=1 THEN GET (0,E,A$,0)
1780  CONVERT K TO C$(###)
1790  A$(1,3)=C$
1800  CONVERT R TO C$(###)
1810  A$(4,6)=C$
1820  IF S1=1 THEN 1840 : REM SKIP OVER ENTERING FILE LABEL
1830  INPUT "Enter internal file label( 38 chrs ) ",A$(7,44)
1840  PUT (0,E,A$,0)
1850  IF S1=1 THEN RETURN
1860  CLOSE (0,E)
1870  #"END OF PROCESSING"
1880  RETURN
```

```
1890  REM PRINT FILE ROUTINE
1900  #" "
1910  INPUT "Options  1. All records  2. Selected records ",O
1920  IF O<1 OR O>2 THEN #"Not a valid option "
1930  IF O<1 OR O>2 THEN 1900
1940  IF O=2 THEN 2110
1950  OPEN (PRINTER,E)
1960  L=0
1970  #"NO.  CHECK  DEPT DIST  AMOUNT"
1980  FOR I=1 TO B
1990    GET (0,E,A$,I)
2000    FOR J=1 TO 14
2010      L=L+1
2020      IF L=K+1 THEN EXIT 2080
2030      M=J*18
2040      #L;"   ";
2050      #A$(M-17,M-13);"   ";A$(M-12,M-11);"   ";A$(M-10,M-8);"   ";
2060      #A$(M-7,M)
2070      NEXT J
2080    NEXT I
2090  CLOSE (PRINTER,E)
2100  GOTO 730
2110  REM
2120  #" "
2130  INPUT "Options  1. By dept.  2. By dist.  3. Both ",O
2140  IF O<1 OR O>3 THEN #"Not valid option number "
2150  IF O=1 THEN INPUT "Enter dept no. ",D$(1,2)
2160  IF O=2 THEN INPUT "Enter dist no. ",D$(4,6)
2170  IF O=3 THEN INPUT "Enter dept. no. ",D$(1,2)
2180  IF O=3 THEN INPUT "Enter dist no. ",D$(4,6)
2190  L=0
2200  OPEN (PRINTER,E)
2210  #"NO.  CHECK  DEPT DIST  AMOUNT"
2220  FOR I=1 TO B
2230    GET (0,E,A$,I)
2240    FOR J=1 TO 14
2250      L=L+1
2260      IF L=K+1 THEN EXIT 2360
2270      M=J*18
2280      IF O=1 THEN D$(4,6)=A$(M-10,M-8)
2290      IF O=2 THEN D$(1,2)=A$(M-12,M-11)
2300      IF D$(4,6)=A$(M-10,M-8) AND D$(1,2)=A$(M-12,M-11) THEN 2320
2310      GOTO 2350
2320      #L;"   ";
2330      #A$(M-17,M-13);"   ";A$(M-12,M-11);"   ";A$(M-10,M-8);"   ";
2340      #A$(M-7,M)
2350      NEXT J
2360    NEXT I
2370  CLOSE (PRINTER,E)
2380  GOTO 730
2390  CLOSE (0,E)
2400  #"End of processing"
```

```
100  FOR J = 1 TO 14
110  M = J*18
120  INPUT"Enter check no. ", A$(M – 17,M – 13)
...
190  NEXT J
```

Example 2.

blocks. This effectively reduced disk storage space requirements while reducing read and write time by 93 percent. To block the records, I used a simple bit of arithmetic (Example 2).

In the FOR statement, the 14 represents the number of records to be blocked together. In the calculation of M, the 18 represents the length of the record. Each pass through the loop moves you 18 bytes through the A string. To demonstrate this, play computer. On the first pass through the loop, J = 1, M ≐ 18. Information entered for the check number will be placed in A string positions 1 through 5 (18-17, 18-13). On the second pass through the loop, J = 2, M = 36, and the check number will be placed in A string positions 19 through 23.

Some additional comments on Business Basic are in order before I turn you loose on the program listing. The CONVERT function used in updating record zero essentially takes a numeric value and converts it to a string in accordance with the mask (###) provided. Include the negative sign and decimal point in the mask if you expect the number to be converted to be negative or have a decimal in it. For BASICs that do not have the CONVERT function, the STR$ and VAL$ commands perform the same function.

The STR$ function will convert numeric information into string information. VAL$ will convert string information into numeric information. Variations of both these commands are available in Business Basic.

Unfortunately, the conversion of numeric information into a string that does not entirely fill the allocated string field results in leading blanks. This does not have any impact until you try to use the VAL$ command to convert the string back into numeric information. At that time you get an "Input error" because the VAL$ command does not evaluate a leading blank. The CONVERT function essentially eliminates this problem by having zero fill any leading blanks. If your BASIC has the STR$ and VAL$ commands, you can duplicate the role the CONVERT function performs.

Conclusion

This ends the commentary on the interpreter and the program. All you need now is to look at the program listing. I hope the REMark statements along the way will adequately describe what is going on in the program. Even if you don't use the program, maybe it will give you some ideas to develop your own.

In Part 2 we will develop the program to use the data files created. ■

✓ Reader Service—see page 179

NEECO

PROUDLY ANNOUNCES THE NEWEST PET MICROCOMPUTERS BY COMMODORE!

PET 2001 — 32K

NEW!

LARGE TYPEWRITER KEYBOARDS NOW AVAILABLE!

The PET™ is now a truly sophisticated Business System with the announcement of these Peripherals.

The PET™, incorporated with the Floppy Disk and Printer makes an ideal business system for most professional and specialized fields: medicine, law, dental, research, engineering, toolmaking, printing, education, energy conservation, etc. . . The PET™ Business System as a management tool, delivers information to all levels of Business, previously attainable only with equipment many times more expensive. The PET™ Business System is one of the most cost efficient business tools today. It offers a wide range of applications from logging management strategy in major corporations to organizing accounts and inventory control of small businesses. Here are just a few of the cost-saving uses in the corporation, professional office or small business: stock control, purchasing, forecasting, manufacturing costing, customer records, mailing lists, etc. . . The PET™, Floppy Disk and Printer, a compatible business system at a reasonable price — Take a closer look at these Peripherals . . .

PRODUCT	DESCRIPTION	PRICE	AVAILABILITY
PET 2001—4K	4K RAM	$ 595	IMMEDIATE
PET 2001—8K	8K RAM	$ 795	IMMEDIATE
PET 2001—16KN (Large Keys)	16K RAM*	$ 995	IMMEDIATE
PET 2001—32KN (Large Keys)	32K RAM	$1295	IMMEDIATE
PET 2023 PRINTER	ROLL FEED	$ 850	IMMEDIATE
PET 2022 PRINTER	TRACTOR/ROLL	$ 995	IMMEDIATE
PET 2040A	SINGLE FLOPPY	$ 895	JUNE/JULY
PET 2040	DUAL FLOPPY*	$1295	IMMEDIATE
PET C2N	2nd Cassette	$ 100	IMMEDIATE

*The 16K/32K (large keyboard) units do not include a cassette drive. Order C2N Cassette. 2040 Floppy Drive requires a 16K or 32K unit. 8K RAM Retrofit available July.

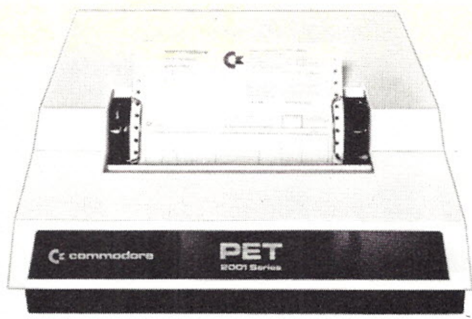
ALL UNITS ARE FULLY TESTED BY NEECO BEFORE SHIPMENT. ALL PET'S ARE WARRANTEED (BY NEECO) FOR 1 FULL YEAR! NEECO IS A FULL CUSTOMER-ORIENTED BUSINESS. PLEASE CALL FOR ADDITIONAL INFORMATION. ALL UNITS ARE IN STOCK & READY TO SHIP. FULL SOFTWARE AVAILABLE!

THE NEW commodore PET PERIPHERALS!

2040

Dual Drive Floppy Disk

The Dual Drive Floppy is the latest in Disk technology with extremely large storage capability and excellent file management. As the Commodore disk is an "Intelligent" peripheral, it uses none of the RAM (user) memory of the PET™. The Floppy Disk operating system used with the PET™ computer enables a program to read or write data in the background while simultaneously transferring data over the IEEE to the PET™. The Floppy Disk is a reliable low cost unit, and is convenient for high speed data transfer. Due to the latest technological advances incorporated in this disk, a total of 360K bytes are available in the two standard 5¼" disks, without the problems of double tracking or double density. This is achieved by the use of two microprocessors and fifteen memory I.C.s built into the disk unit. Only two connections are necessary — an A/C cord and PET™ interface cord.



2022

Tractor Feed Printer

The Tractor Feed Printer is a high specification printer that can print onto paper (multiple copies) all the PET™ characters — letters (upper and lower case), numbers and graphics available in the PET™. The tractor feed capability has the advantage of accepting mailing labels, using standard preprinted forms (customized), cheque printing for salaries, payables, etc. Again, the only connections required are an A/C cord and PET™ connecting cord. The PET™ is programmable, allowing the printer to format print for: width, decimal position, leading and trailing zero's, left margin justified, lines per page, etc. It accepts 8½" paper giving up to four copies. Bidirectional printing enables increased speed of printing.

CALL FOR COMPLETE SPECIFICATIONS! IMMEDIATE DELIVERY!

NEW ENGLAND AREA SUB-DEALERSHIP INQUIRIES INVITED!

NEW IBM SELECTRIC II

TYPEWRITERS FOR PET!
(BRAND NEW IBM II)

UNIT INTERFACES VIA THE USER PORT. FOR THE FIRST TIME A BRAND NEW IBM SELECTRIC IS AVAILABLE FOR YOUR PET! FULL CASSETTE OR DISK WORD PROCESSING PACKAGES. CALL FOR MORE INFO. PET - IBM $1995

BUSINESS SOFTWARE

BUSINESS PACKAGES AVAILABLE ON THE 2040 DUAL FLOPPY:
BUSINESS PAYROLL —
COMPLETE MAILING LIST —
STATISTICAL PACKAGE —
GENERAL LEDGER —
(UP TO 1,024 A/R's, 512 A/P's, 512 GEN LED ACCTS, POSTING, DISBURSEMENT, AGEING, B/S, I?S, AND MANY OTHER FUNCTIONS!) COMPLETE WORD PROCESSING PACKAGES.

NEECO NOW OFFERS A FULL ONE YEAR WARRANTY ON ALL PET'S SOLD BY NEECO! OUR NEW JUNE CATALOGS ARE NOW AVAILABLE — ARE YOU ON OUR MAILING LIST? NEECO OFFERS FULL SALES, SYSTEMS, AND SERVICE SUPPORT FOR ALL OF OUR CUSTOMERS — WHY NOT BUY FROM THE BEST? DEALER INQUIRIES INVITED.

—NEECO OFFERS COMPLETE SOFTWARE (COMMERCIAL & CONSUMER)

NEECO

NEW ENGLAND ELECTRONICS CO., INC.
679 HIGHLAND AVE., NEEDHAM, MASS. 02194
MON. - FRI. 9:30 - 5:30, EST.

(617) 449-1760
MASTERCHARGE OR VISA ACCEPTED
TELEX NUMBER 951021, NEECO

Computer Careers in Carolina

"It feels pretty good to be number one in vocational electronics in North Carolina."

James A. Gupton, Jr.
7416-G Pebblestone Dr.
Charlotte NC 28212

Across the road from the cemetery, behind a field of soybeans, stands the Union County Career Education Center, a new concept for high-school vocational education in North Carolina.

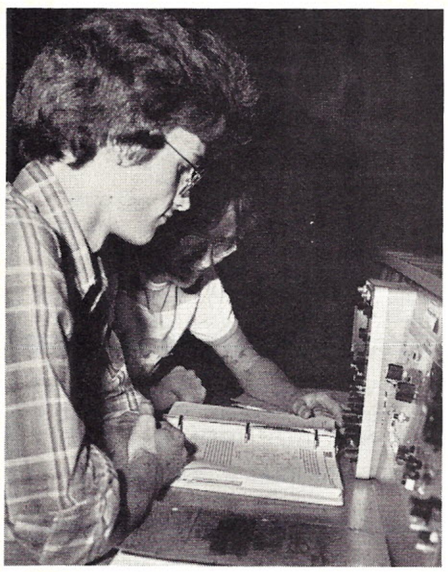
Of course, vocational electronics is not new at the high-school level. What makes the Union County Career Center's vocational-electronics program unique is the number of firsts it has accomplished and the level of its program.

On the Honor Roll

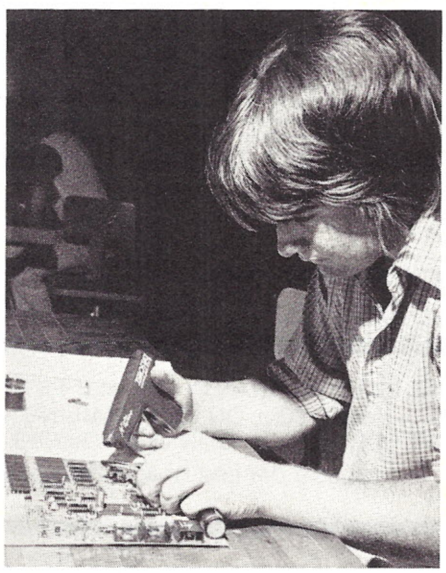
The Career Center was the first in the United States to fully integrate the entire Heath Company's Continuing Education courses into the vocational-electronics curriculum. It is an adult-level program where the students are taught as adults. Each student wears a name badge with his name prefixed by the title Mr. (for example, Mr. J. Beardsly). All communication between student and teacher is on a formal basis. Each student is schooled in the adult responsibilities of industrial employment and trained to perform at professional levels . . . an unusual accomplishment in a time when there is so much publicity on high-school graduates who cannot read.

Suppose we take a look at some of the unusual "firsts" achieved by the Union County Career Center's vocational-electronics program.

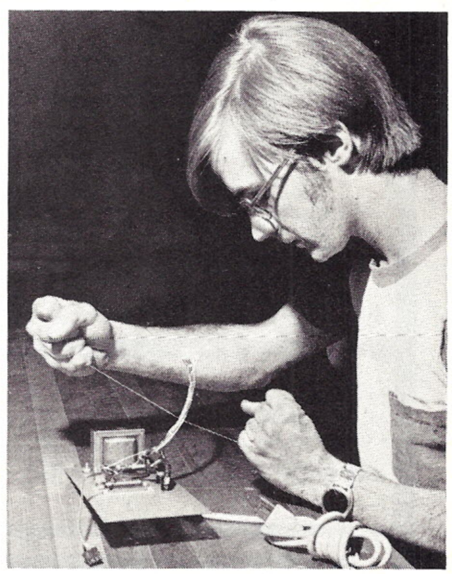
- First high school in North Carolina to establish an electronics program on an adult level.
- First high school in the United States to provide hands-on experience on the Radio Shack TRS-80 microcomputer (September 1977).
- First high school in the United States to give hands-on training in assembly of the Heath GRS-2001 Color Television Receiver and the Heath H8 microcomputer.
- First high school in the United States to include microcomputer and BASIC-language programming in vocational electronics (1977).
- First high school in the United States to have students assemble the Terrapin Turtle, a computer-controlled robot.



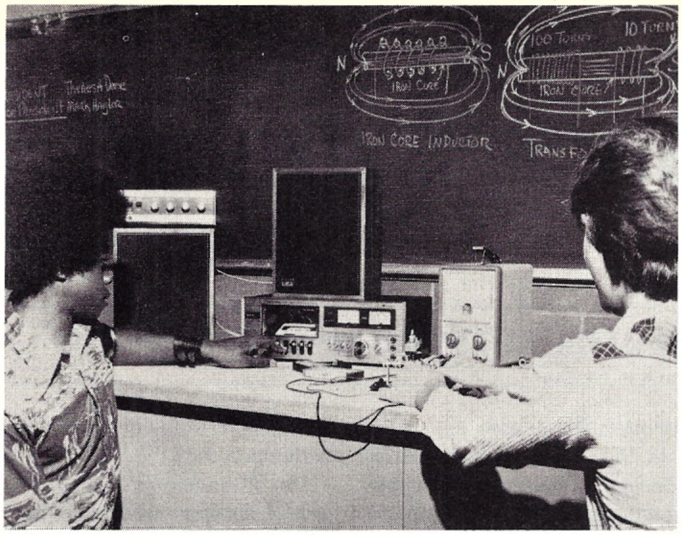
Tommy Stegall and Bennie Horn learn about direct current through the Heath Continuing Education Electronic course.



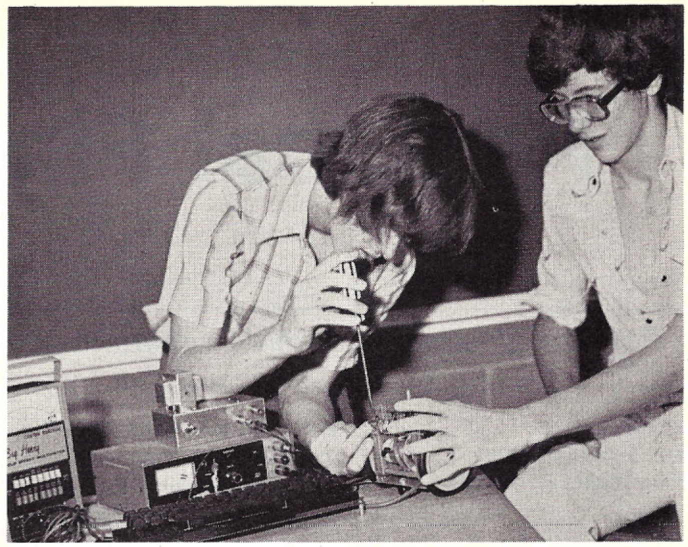
Michael Helms eliminates all static charges on the main PC board and MOS ICs with the Zerostat ion gun.



Douglas Knight demonstrates his skill in lacing the Altair 680b's power supply connection to the motherboard.



Instructor-recorder cassette tapes add new emphasis for Dale O'Leary (left) and Eddie Fincher (right) on understanding the basic fundamentals of electronics at the Career Center.



Richard Voss and Jeff Dunn add the finishing touches to the assembly of the Terrapin Turtle.

Course Description

Wouldn't you have welcomed the opportunity to attend an electronics class in a high school that offered a program such as this?

First quarter—Direct-current fundamentals and laboratory practice. Manual-skill training in the proper use of hand tools, wire-wrapping techniques and soldering techniques for printed circuits.

Second quarter—Alternating-current fundamentals and laboratory practice. Training in the use of power supplies, rectifiers, meters, oscilloscopes, decibel meters. Manual-skill training in metal working, drilling and tapping metals.

Third quarter—Semiconductors and electro-optic devices and laboratory practice. Training in electronic drafting, transistor operation, circuit breadboarding. Character building in leadership, citizenship and how to apply for electronic employment.

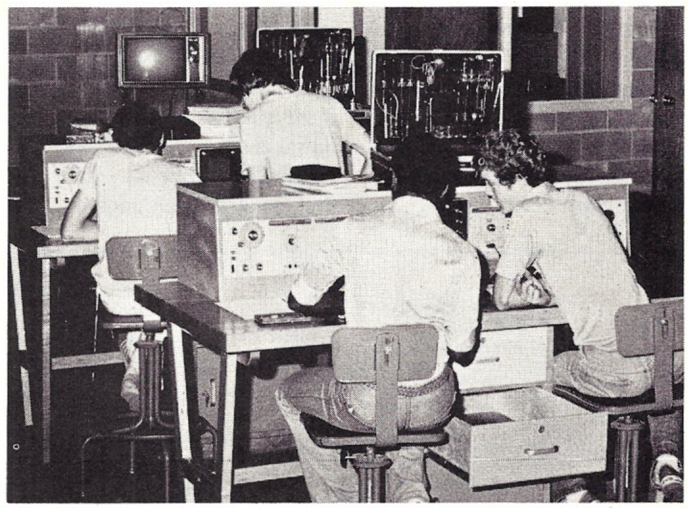
Fourth quarter—Student research and development project. The student must select an electronics project, breadboard the circuit, design a printed circuit board, etch the PC board and assemble the project. The total project is then prepared as a written R&D project on the same requirements as engineers at ITT prepare their R&D project reporting.

Fifth quarter—Electronic circuit specialization and laboratory practice. Here the student works with specialized electronic circuits until he or she can readily identify that specific circuit anywhere and fully identify how it operates. This includes oscillators, amplifiers of all types, multivibrators, flip-flops, regulators, etc.

Sixth quarter—Digital logic circuits. Here the student masters the complexities of integrated DIP logic circuits, the different types of gate functions and the basic computer logic applications.

Seventh and eighth quarters—Student option: A. Microcomputer circuit and design, B. BASIC language programming for computers. The students continue the Heath Continuing Education courses for the option they have selected. Upon mastery of the Heath course, they then apply this training to actual microcomputers in the class. Programming offers Extended BASIC and TRS-80 Level II. Computers include the TRS-80 and the Heath H8.

A major student-motivation factor is the amount of local and national publicity, through photographs, the students-in-training receive. At the beginning of each school year, each student is asked to sign publication releases so that his photograph can be included in articles on the electronics program. Since 1977, these students have appeared in four articles: in Mits/Pertec's *Computer Notes*, Radio Shack's in-house organ, *Intercom, Elementary Electronics* and a new book on microcomputers, *Getting Down to Business with Microcomputers.*



The UCCC students work in teams of two at each laboratory bench. VACO Electronc Tool kits provide quality and professional hand-tool training for each student.

How did such a vocational-electronics program develop in the first place? You must admit that it is quite unusual for a high-school level program. It was a case of desperation and lack of sufficient funds! The initial training equipment specified by the Career Center's architect provided the laboratory benches, panel meters and power supplies, but only enough electronic experimental aids for the first semester. To purchase the same line of training equipment for the remaining three semesters would have cost over $9600, and less than $2000 remained in the vocational electronics initial equipment budget. Only Heath Company could supply the required training equipment and materials at a cost within the budget's remaining funds.

To realize that an architect's oversight made possible this unique vocational-electronics program is almost frightening . . . what if there had been no oversight? Then what type of course would the Career Center have today? Perhaps only radio and television repair. ■

Personal Finance System

Part 1 of this series covered accounts payable. This installment takes in receivables.

James McClure
1019 Van Kirk Street
Philadelphia PA 19149

Well, Jimmy, here it is—all 784 billion. Now we can get some more of those planes you liked and still have money left over for putting in some nuclear reactors. But wait a minute, we've got to take out 13.4 billion for Midwestern food subsidies, and don't forget we promised the military an extra 25 billion this year. Let's see, that leaves us with 745.6 billion, less the 12 million for new jobs and another 65 billion for increased welfare benefits and 33.3 million for . . ."

Like everyone else, President Carter is having money problems. While the income of the average citizen may be "just peanuts" compared to the three-quarter-trillion-dollar national budget, it puts food on the table, and so any scheme that may stretch income deserves a closer look.

```
        REM ***************
5000    REM PAY A BILL
        REM ***************
        INPUT "ACCOUNT NUMBER?";ACCOUNT.NO
        OPEN "CURRENT.DAT" RECL 72 AS 1
        RECORD=ACCOUNT.NO+1
        GOSUB 10000      REM READ RECORD
        PRINT USING F3$;ACCOUNT.NO,ACCOUNT$
        PRINT USING "BALANCE DUE $$####.##";BALANCE
        INPUT "AMOUNT OF PAYMENT?";PAYMENT
        INPUT "PAYMENT BY CHECK?";RES$
        IF LEFT$(RES$,1) EQ "N" THEN \
                CHECK.NO=0 :\
                INPUT "DATE OF PAYMENT?";CHECK.DATE$ :\
                GOTO 5100
*       OPEN "BANK$ACC.DAT" RECL 50 AS 2
*       READ # 2,2;C.BALANCE,LAST.DATE$
*       IF C.BALANCE-PAYMENT LT 0 THEN \
*               CLOSE 2 :\
*               GOTO 2990
*       PRINT # 2,2;C.BALANCE-PAYMENT,DATE$
*       CLOSE 2

        ... REST OF ROUTINE IS UNCHANGED

        REM ***************
8000    REM WRITE A CHECK FOR CASH OR MISC
        REM ***************
        INPUT "FOR CASH OR MISCELLANEOUS?";ACCOUNT$
        IF LEFT$(ACCOUNT$,1) EQ "C" THEN \
                ACCOUNT=0:ACCOUNT$="CASH" \
        ELSE \
                ACCOUNT=-1
        IF ACCOUNT=-1 THEN \
                INPUT "CHECK FOR WHAT?";ACCOUNT$
        INPUT "AMOUNT OF CHECK?";TOTAL.DUE
*       OPEN "BANK$ACC.DAT" RECL 50 AS 1
*       READ # 1,2;C.BALANCE,LAST.DATE$
*       IF C.BALANCE-TOTAL.DUE LT 0 THEN 2990
*       PRINT # 1,2;C.BALANCE-TOTAL.DUE,DATE$
*       CLOSE 1

        ... REST OF ROUTINE IS UNCHANGED
```

Fig. 1. Modifications to accounts-payable program (lines inserted are marked with asterisks).

In Part 1 of this finance package, I presented a program that would allow a properly equipped computer to keep track of checks, bills and other payments, and ultimately produce an accurate record of family spending trends.

In this article, I want to concentrate on income such as that received from a full-time job, tax refunds (if you believe in miracles) and other miscellaneous sources. Close scrutiny of spending and income may result in better control of your money, and I hope this control will help you save some of the fleeing green. If you find yourself watching your hard-earned dollars disappear into a million hungry coffers, none of which are yours, then this program is for you. If not, then consider yourself lucky and go play Star Trek.

As mentioned earlier, the purpose of the program is to keep track of all income. Generally, income can be divided into two kinds: that from a full-time job and that from odd jobs and miscellaneous sources. As a result, the program maintains two data files: "Pay$stubs.dat," which contains tax information from each paycheck, and "Bank$acc.dat," which contains a list of miscellaneous deposits.

As presented, the program accepts pay stub data in Philadelphia School District format. The data available on such a stub can be broken into two areas: string data and numeric data. For example, the code number of the stub, since it consists of a series of letters and/or digits, is string data, whereas pay totals, amounts of deductions, etc., are examples of numeric data.

The Philadelphia education format incorporates three string fields and 23 numeric fields of useful data; therefore, the first two lines of the parameter block (starting at line 101) assign the values 3 and 23, respectively, to the variables "string.fields" and "numeric.fields." These two assignments should be changed to reflect the data fields present on your pay stub.

Next, field 8 is designated to hold the gross pay amount by the assignment of 8 to the variable "gross.pay." Field 23 is designated to hold the net pay figure by the assignment of 23 to the variable "net.pay." These two lines must be changed to assign the correct field numbers to the variables "gross.pay" and "net.pay."

Finally, the names of the fields themselves must be entered into the data statements of the parameter block. Note that they are present in a specific order (the order in which they appear on the pay stub) and that the string field titles precede the numeric titles. This must be done for simplicity; otherwise, the computer will not know what type of data it is handling at a given moment.

The remaining changes are fairly minor. The routine starting at line 10000 reads pay stub records from the disk into two arrays: array R$, which holds the string data, and array R, which holds the numeric data. Each element of the arrays constitutes a field from the pay stub. The fields must be read in order, starting with element one of string array R$ and ending with the last element of numeric array R (which in this

case is R(23)).

Looking at the code of routine 10000, you might think that a FOR loop would be much more efficient; however, this will not work (at least not with CBASIC). All the fields of a record must be read with one statement. Therefore, you must change the read statement so that the proper number of string and numeric fields is input.

In essence, the program emulates a personal bank and has a checking and a savings account. The computer maintains a balance for each account, which is updated as deposits and withdrawals are made. If all information is entered carefully, you will always know the balance of both accounts, thereby avoiding embarrassing and costly overdrafts. Necessary information regarding paychecks is also kept for tax purposes and is easily accessible so that April 14, 11:59 PM crisis may be avoided.

Requirements

Ideally, the program should be run on a system with 28K (or more) memory, a video terminal, one or more disk drives and

```
        REM ***************
2500    REM SAVINGS ACCOUNT WITHDRAWL
        REM ***************
        INPUT "AMOUNT OF WITHDRAWL?";AMOUNT
        OPEN "BANK$ACC.DAT" RECL 50 AS 1
        READ # 1,3;C.BALANCE,LAST.DATE$
        IF C.BALANCE-AMOUNT LT 0 THEN 2990
        PRINT # 1,3;C.BALANCE-AMOUNT,DATE$
        CLOSE 1
        OPEN "HISTORY.DAT" RECL 72 AS 1
        READ # 1,1;RECORD
        INPUT "MONEY FOR WHAT?";ACCOUNT$
        INPUT "DATE OF WITHDRAWL?";DUE.DATE$
        IF LEN(DUE.DATE$) EQ 0 THEN DUE.DATE$=DATE$
        ACCOUNT=-2
        BALANCE=0
        TOTAL.DUE=AMOUNT
        GOSUB 11000      REM WRITE RECORD
        PRINT # 1,1;RECORD+1
        CLOSE 1
        PRINT USING "NEW BALANCE IS $$######.##";C.BALANCE-AMOUNT
        RETURN
2990    PRINT "ACCOUNT OVERDRAWN"
        PRINT USING "CURRENT BALANCE $$######.##";C.BALANCE
        CLOSE 1
        RETURN
```

Fig. 2. Additions to accounts payable.

```
        THE FOLLOWING CODE IS FROM THE BLOCK AT LINE 200
                 OF THE ACCOUNTS-PAYABLE PROGRAM
     PRINT "G=PRINT BILLS DUE THIS MONTH";TAB(40); \
          "H=WRITE CHECK (NOT FOR BILL)"
*    PRINT "I=EDIT AN ACCOUNT";TAB(40);"J=SAVINGS ACCOUNT WITHDRAWL"
*    PRINT "K=RETURN TO DISK OPERATING SYSTEM"
     PRINT:PRINT:PRINT:PRINT
300  INPUT "OPTION (?=LIST OPTIONS)?";OPTION$
     IF OPTION$="?" OR LEN(OPTION$) EQ 0 THEN 200
     OPTION=ASC(OPTION$)-64
*    IF OPTION LT 1 OR OPTION GT 11 THEN 300
     ON OPTION GOSUB 2000,1000,3000,4000,5000, \
*         6000,7000,8000,8100,2500,8999

     ... REST OF CODE IS THE SAME
```

Fig. 3. Implementation of the new routine (lines which have been changed are marked with asterisks).

Program listing.

```
        REM ***************
100     REM PERSONAL ACCOUNT SYSTEM
        REM
        REM JAMES MCCLURE
        REM JUNE 1978
        REM ***************
        REM
        REM *****************************************
        REM ACCOUNTS RECEIVABLE SYSTEM
        REM *****************************************
        YES=-1:NO=0:EXIT=1
        F1$="/......../"
        F2$="######.##"
        F4$="/....../ /....../ $$######.## $$######.## $$#####.##"+ \
             "      ##.#%"
        H4$="  CODE      DATE      GROSS PAY      NET PAY     TOTAL DED"+ \
             "    PERCENT"
        F5$=" !   /................./$$######.##     /....../ "
101     REM ********** PARAMETER BLOCK **********
        REM IDENTIFY DATA FIELDS ON PAY STUB
        REM STRING FIELDS ONE AND TWO MUST CONTAIN THE PAY STUB CODE
        REM AND DATE RESPECTIVELY!
        STRING.FIELDS=3 REM NUMBER OF STRING DATA FIELDS
        NUMERIC.FIELDS=23 REM NUMBER OF NUMERIC DATA FIELDS
        GROSS.PAY=8 REM NUMERIC FIELD CONTAINING GROSS PAY
        NET.PAY=23 REM NUMERIC FIELD CONTAINING NET PAY
        DIM R(NUMERIC.FIELDS),R$(STRING.FIELDS),T(NUMERIC.FIELDS)
        REM FIELD HEADINGS
        DATA "STUB CODE","CHECK DATE","PERIOD ENDING","BI-WEEKLY"
        DATA "P.I.ABS.DED.","OTHER ABS.DED.","BASE EARNINGS"
        DATA "P.I.ABS. DAYS","P.I.ABS. PAY","PERS. LEAVE"
        DATA "GROSS EARNINGS","FED.TAXABLE EARNS.","FED.TAX"
        DATA "SOC.SEC.EARNS.","SOC.SEC.TAX"
        DATA "CITY TAX","STATE DED.","UNION DED.","H.I.PREMIUM"
        DATA "RET.EARNS.","RET.DED.","HEALTH PLAN DED.","BLUE CROSS"
        DATA "MAJOR MED.","MED.SURGICAL","NET PAY"
        REM THE NEXT FEW LINES CHECK TO SEE IF THE FILE
        REM "PAY$STUB.DAT" EXISTS. IF NOT, IT AND THE FILE
        REM "BANK$ACC.DAT" ARE CREATED AND INITIALIZED TO EMPTY.
        IF END # 1 THEN 110
        OPEN "PAY$STUB.DAT" RECL 50 AS 1
        GOTO 120
110     CREATE "PAY$STUB.DAT" RECL 250 AS 1
        PRINT # 1,1;2
        CLOSE 1
        CREATE "BANK$ACC.DAT" RECL 50 AS 1
        PRINT # 1,1;4
        PRINT # 1,2;0," "
        PRINT # 1,3;0," "
120     CLOSE 1
        INPUT "TODAY'S DATE?";DATE$
130     PRINT
        PRINT "ACCOUNTS RECEIVABLE SYSTEM";TAB(50);"DATE: ";DATE$
        PRINT:PRINT:PRINT:PRINT
        PRINT "ALLOWABLE OPTIONS ARE:"
        PRINT
        PRINT "A=ENTER NEW PAY CHECK";TAB(40);"B=LIST PAY CHECKS"
        PRINT "C=PRINT A SPECIFIC PAY CHECK";TAB(40);
        PRINT "D=DEPOSIT AN EXISTING PAY CHECK"
        PRINT "E=ENTER MISC DEPOSIT";TAB(40);"F=LIST ALL DEPOSITS"
        PRINT "G=PRINT ACCOUNT BALANCES";TAB(40);"H=EDIT A RECORD"
        PRINT "I=RETURN TO OPERATING SYSTEM"
        PRINT:PRINT:PRINT:PRINT
140     INPUT "OPTION (?=LIST OPTIONS)?";OPTION$
        PRINT FRE
        IF OPTION$="?" OR LEN(OPTION$) EQ 0 THEN 130
        OPTION=ASC(OPTION$)-64
        IF OPTION LT 1 OR OPTION GT 9 THEN 140
        ON OPTION GOSUB 1000,2000,3000,4000,5000,6000,7000,8000,8999
        CONSOLE
        PRINT
```

Field heading	Array element	Edit field
Pay stub code[1]	R$(1)	1
Check date[1]	R$(2)	2
Period ending	R$(3)	3
Biweekly pay	R(1)	4
Personal illness absence deduction	R(2)	5
Other absences deduction	R(3)	6
Base earnings	R(4)	7
Personal illness absence days	R(5)	8
Personal illness absence pay	R(6)	9
Personal leave	R(7)	10
Gross earnings	R(8)	11
Federal taxable earnings	R(9)	12
Federal tax	R(10)	13
Social security earnings	R(11)	14
Social security tax	R(12)	15
City tax	R(13)	16
State deductions	R(14)	17
Union deductions	R(15)	18
Health insurance premium	R(16)	19
Retirement earnings	R(17)	20
Retirement deduction	R(18)	21
Health plan deduction	R(19)	22
Blue Cross	R(20)	23
Major medical	R(21)	24
Medical surgical	R(22)	25
Net pay	R(23)	26

Note 1. Pay stub code and check date must occupy these spots.

Fig. 4. Pay stub record.

some form of printer. Software required is the same for this and the accounts-payable package: CP/M and CBASIC, both obtainable from Digital Research. Those unfamiliar with the CP/M disk operating system would do well to read "CP/M Primer" by Dr. John F. Stewart, p. 30, in the April 1978 issue of *Kilobaud*.

The accounts-payable package, presented in Part 1, was designed to run by itself, as well as in combination with the other two programs of the series. However, the accounts-receivable program is not independent; it requires the data collected by accounts payable.

In order for the two programs to communicate, several additions must be made to the first program (orginally left out to allow it to be independent). These additions are summarized in Figs. 1 through 3 and should take only a few minutes to complete.

Several lines of code are added to the routines at lines 5000 and 8000, and a new function is added at line 2500. The added lines automatically deduct checks written for bills or miscellaneous purposes from the checking account. The new function at line 2500, J, allows withdrawals to be made from the savings account. Both of these additions check for overdraft conditions and will not complete the transaction if such a condition exists.

Modifications

Due to the specific nature of pay stubs, changes will have to be made in the program before it can be used, even if you are running it on the recommended hardware and software. Basically, four blocks of the program need to be modified: the parameter block and three I/O subroutines.

The subroutine at line 11000 writes pay stub records to the disk and is identical to the subroutine at 10000, except that it is a print rather than a read operation. Thus, when subroutine 10000 has been properly modified, it may be copied over to line 11000, changing the word "read" to "print."

Finally, the subroutine starting at line 11500 will need to be modified. This routine writes pay stub records to the console and is functionally equivalent to subroutine 11000, except that it writes to the CRT. Again, the data is output in order, with several items to each line, and the format may be changed to suit your aesthetic taste.

Although these modifications take many paragraphs to explain, they are easy to implement, and I suspect that someone reasonably experienced with BASIC should be able to make the changes in about an

```
        GOTO 140
        REM ***************
1000    REM INPUT NEW PAY CHECK
        REM ***************
        PRINT "INPUT NEW PAY CHECK";
        GOSUB 9100              REM MAKE SURE
        IF ANS EQ EXIT OR ANS EQ NO THEN 1099
        PRINT
        GOSUB 10500             REM READ DATA FROM CRT
        OPEN "PAY$STUB.DAT" RECL 250 AS 1
        READ # 1,1;RECORD       REM FIND NEXT AVAILABLE RECORD
        GOSUB 11000             REM WRITE DATA TO DISK
        PRINT # 1,1;RECORD+1
        CLOSE 1
        PRINT "DEPOSIT CHECK NOW";
        GOSUB 9100              REM GET ANSWER
        IF ANS EQ NO THEN GOTO 1099
        GOSUB 1500
1099    RETURN
        REM ***** DEPOSIT SUBROUTINE *****
1500    S.AMOUNT=0
        C.AMOUNT=R(NET.PAY)
        PRINT "FULL AMOUNT DEPOSITED";
        GOSUB 9100              REM GET ANSWER
        IF ANS EQ YES THEN 1600
        INPUT "WHAT AMOUNT TO CHECKING ACCOUNT?";C.AMOUNT
        INPUT "WHAT AMOUNT TO SAVINGS ACCOUNT?";S.AMOUNT
1600    INPUT "WHAT DATE CHECK DEPOSITED?";DATE.DEP$
        IF LEN(DATE.DEP$) EQ 0 THEN DATE.DEP$=DATE$
        IF S.AMOUNT+C.AMOUNT EQ 0 THEN 1900
        OPEN "BANK$ACC.DAT" RECL 50 AS 1
        READ # 1,1;RECORD
        IF C.AMOUNT EQ 0 THEN 1700
        ACC$="C"
        ACCOUNT$="PAY STUB "+R$(1)
        AMOUNT=C.AMOUNT
        GOSUB 11100             REM WRITE DATA TO DISK
        READ # 1,2;BALANCE,LAST.DATE$
        PRINT # 1,2;BALANCE+AMOUNT,DATE$
        RECORD=RECORD+1
        PRINT # 1,1;RECORD
1700    IF S.AMOUNT EQ 0 THEN 1800
        ACC$="S"
        ACCOUNT$="PAY STUB "+R$(1)
        AMOUNT=S.AMOUNT
        GOSUB 11100             REM WRITE DATA TO DISK
        READ # 1,3;BALANCE,LAST.DATE$
        PRINT # 1,3;BALANCE+AMOUNT,DATE$
        PRINT # 1,1;RECORD+1
1800    CLOSE 1
1900    RETURN
        REM ***************
2000    REM SHORT FORM LIST
        REM ***************
        PRINT "ANY SPECIFIC MONTH/YEAR";
        GOSUB 9100              REM GET ANSWER
        IF ANS EQ EXIT THEN 2099
        IF ANS EQ YES THEN \
                INPUT "WHAT MONTH/YEAR?";MON$ :\
                YR$=RIGHT$(MON$,2):MON$=LEFT$(MON$,2) \
        ELSE \
                MON$=""
        GOSUB 9000
        OPEN "PAY$STUB.DAT" RECL 250 AS 1
        READ # 1,1;RECORDS
        PRINT H4$
        PRINT
        FOR RECORD=2 TO RECORDS-1
                GOSUB 10000     REM READ RECORD
                IF (LEFT$(R$(2),2) EQ MON$ AND RIGHT$(R$(2),2) EQ YR$)
                        OR LEN(MON$) EQ 0 THEN \
```

hour. Of utmost importance to the modification process is the format chosen for the pay stub records; therefore, a sample format (used in the preparation of this program) is shown in Fig. 4.

Operation

The program is written to be operated by both technical and nontechnical persons, so it incorporates a numer of error-reducing features. For example, if the wrong function is accidentally requested, the user may return to the command mode by typing "exit." Furthermore, all answers to yes/no questions are checked for appropriateness, and a special message is printed if the answer is other than a yes or no.

The system allows abbreviations wherever they are feasible. Generally, if an answer is requested, it is sufficient to type only the first letter of the response. This greatly reduces redundant typing, and with it, the chance for errors.

The program provides nine functions, any of which may be selected by typing the appropriate letter followed by a carriage return. The computer will prompt for all necessary information.

When the program is first executed, it will create several data files and initialize them to empty. Afterward, a menu is printed at the console, listing all available options.

The first function, A, allows paychecks to be entered as they are received. The computer will prompt for each field on the pay stub in order, allowing you to enter the proper information. No decimals are necessary when numbers are being entered.

After all the fields have been input, the computer will ask whether you wish to deposit the paycheck into the checking (or savings) account. If you answer yes, it will ask if the full amount is to be deposited into the checking account, and if not, exactly what amounts are to go to the savings and/or checking accounts. If you do not deposit the check immediately, this may be done at a later date using the D option. This latter method may be useful if you are entering the pay stub information into the computer before you have actually taken the check to the bank.

Function B prints a short form list of all paychecks and is handy for inducing a heart attack. The computer prints the code number of each check, followed by its date, gross and net pay fields. In the fifth column, the difference between the gross and net pay fields is printed. This difference represents the total deductions made from your pay by the city, state, etc. If this column doesn't shake the old ticker, the last one will; it shows the percentage of your gross pay lost to deductions and can be quite a revelation.

The third function, C, allows individual pay stubs to be ex-

```
A>CRUN ACC/REC
CRUN VER 1.02

TODAY'S DATE? 06/09/78

ACCOUNTS RECEIVABLE SYSTEM                              DATE: 06/09/78

ALLOWABLE OPTIONS ARE:

A=ENTER NEW PAY CHECK                 B=LIST PAY CHECKS
C=PRINT A SPECIFIC PAY CHECK          D=DEPOSIT AN EXISTING PAY CHECK
E=ENTER MISC DEPOSIT                  F=LIST ALL DEPOSITS
G=PRINT ACCOUNT BALANCES              H=EDIT A RECORD
I=RETURN TO OPERATING SYSTEM
```

Sample 1. Initialization dialogue.

```
        DED=R(GROSS.PAY)-R(NET.PAY):DEDP=100*(DED/R(GROSS.PAY)) :\
        PRINT USING F4$;R$(1),R$(2),R(GROSS.PAY),R(NET.PAY),DED,DEDP
        NEXT RECORD
        CLOSE 1
2099    RETURN
        REM ***************
3000    REM PRINT OUT A SPECIFIC PAY STUB
        REM ***************
        INPUT "PAY STUB CODE?";CODE$
        IF CODE$ EQ "EXIT" THEN 3099
        GOSUB 9000
        GOSUB 3500              REM SEARCH ROUTINE
        IF FAIL EQ YES THEN 3099
        GOSUB 10000             REM READ RECORD
        GOSUB 11500             REM PRINT RECORD
        CLOSE 1
3099    RETURN
3500    REM ***** SEARCH SUBROUTINE *****
        OPEN "PAY$STUB.DAT" RECL 250 AS 1
        READ # 1,1;RECORDS
        FOR RECORD=2 TO RECORDS-1
                REM THE FOLLOWING LINE READS ONLY THE FIRST ELEMENT
                REM OF EACH RECORD...THAT IS, THE PAY STUB CODE
                READ # 1,RECORD;R$(1)
                IF CODE$ EQ R$(1) THEN \
                        FAIL=NO :\
                        GOTO 3599
        NEXT RECORD
        PRINT "NO SUCH PAY STUB"
        FAIL=YES
        CLOSE 1
3599    RETURN
        REM ***************
4000    REM DEPOSIT EXISTING PAY STUB
        REM ***************
        INPUT "STUB CODE?";CODE$
        IF CODE$ EQ "EXIT" THEN 4099
        GOSUB 3500              REM SEARCH ROUTINE
        IF FAIL EQ YES THEN 4099
        GOSUB 10000             REM READ ENTIRE RECORD
        CLOSE 1
        PRINT USING "PAY CHECK VALUE $$#####.##";R(NET.PAY)
        GOSUB 1500              REM DEPOSIT SUBROUTINE
4099    RETURN
        REM ***************
5000    REM MAKE DEPOSIT
        REM ***************
        RECORD=0
        INPUT "CHECKING OR SAVINGS ACCOUNT?";RES$
        IF RES$ EQ "EXIT" THEN 5099
        IF LEFT$(RES$,1) EQ "C" THEN RECORD=2
        IF LEFT$(RES$,1) EQ "S" THEN RECORD=3
        IF RECORD EQ 0 THEN 5000
        INPUT "AMOUNT OF DEPOSIT?";AMOUNT
        INPUT "MONEY FROM WHOM?";ACCOUNT$
        INPUT "DATE DEPOSITED?";DATE.DEP$
        IF LEN(DATE.DEP$) EQ 0 THEN DATE.DEP$=DATE$
        ACC$=LEFT$(RES$,1)
        OPEN "BANK$ACC.DAT" RECL 50 AS 1
        READ # 1,RECORD;BALANCE,LAST.DATE$
        PRINT # 1,RECORD;BALANCE+AMOUNT,DATE$
        READ # 1,1;RECORDS
        RECORD=RECORDS
        GOSUB 11100             REM WRITE DATA TO DISK
        PRINT # 1,1;RECORDS+1
        CLOSE 1
5099    RETURN
        REM ***************
6000    REM LIST DEPOSITS
        REM ***************
        TOTAL=0
```

```
OPTION (?=LIST OPTIONS)? A
INPUT NEW PAY CHECK? Y

NO DECIMALS PLEASE

STUB CODE? A518212
CHECK DATE? 05/26/78
PERIOD ENDING? 05/19/77              *** ERROR - SHOULD BE 05/19/78 ***
BI-WEEKLY? 115507
P.I.ABS.DED.? 0
OTHER ABS.DED.? 0
BASE EARNINGS? 115507
P.I.ABS. DAYS? 0
P.I.ABS. PAY? 0
PERS. LEAVE? 0
GROSS EARNINGS? 115507
FED.TAXABLE EARNS.? 115507
FED.TAX? 23966
SOC.SEC.EARNS.? 115507
SOC.SEC.TAX? 6988
CITY TAX? 4981
STATE DED.? 2541
UNION DED.? 1155
H.I.PREMIUM? 1444
RET.EARNS.? 115507
RET.DED.? 6064
HEALTH PLAN DED.? 0
BLUE CROSS? 0
MAJOR MED.? 0
MED.SURGICAL? 0
NET PAY? 68368

DEPOSIT CHECK NOW? N

OPTION (?=LIST OPTIONS)? B
ANY SPECIFIC MONTH/YEAR? N
PRINT ON TTY? Y

  CODE      DATE       GROSS PAY      NET PAY    TOTAL DED    PERCENT

A518212  05/26/78     $1155.07       $683.68     $471.39      40.8%
```

Sample 2. Functions A and B.

amined. This is useful to see if all information has been entered correctly.

As I mentioned earlier, function D will permit the deposit of previously entered paychecks. For instance, if you received the check Friday but didn't deposit it at the bank until Monday, then function D may be used to record this deposit.

Occasionally, you may receive income from a source other than your regular job. These miscellaneous deposits can be made with the E option. After instructing the computer as to which account (checking or savings) the check is to be deposited, you will be requested to enter the amount, source and date of the deposit. As in the accounts-payable program, typing a space will insert the current date.

Once these miscellaneous deposits have been entered, they may be listed using the F option. The source, amount and date of each deposit will be printed, followed by a total.

To examine the balance of either the checking or savings account, use the G option. The computer will print the balance of each account, along with the dates of the last transactions involving the accounts.

It is inevitable that errors will be made while data for the pay stub or deposit records is being entered. To correct these mistakes, option H has been provided. Invoking this option will allow any record within the sys-

```
        PRINT "ANY SPECIFIC ACCOUNT";
        GOSUB 9100              REM GET ANSWER
        IF ANS EQ EXIT THEN 6099
        IF ANS EQ YES THEN \
                INPUT "CHECKING OR SAVING?";ACC1$ \
        ELSE \
                ACC1$=""
        GOSUB 9000
        OPEN "BANK$ACC.DAT" RECL 50 AS 1
        READ # 1,1;RECORDS
        PRINT
        PRINT "ACN          ACCOUNT                  AMOUNT     DATE DEP"
        PRINT "---  --------------------  ----------  ----------"
        PRINT
        FOR RECORD=4 TO RECORDS-1
                GOSUB 10100         REM READ DATA FROM DISK
                IF LEN(ACC1$) EQ 0 OR LEFT$(ACC1$,1) EQ ACC$ THEN \
                PRINT USING F5$;ACC$,ACCOUNT$,AMOUNT,DATE.DEP$ :\
                TOTAL=TOTAL+AMOUNT
        NEXT RECORD
        PRINT
        PRINT "-----------------------------------------------"
        PRINT TAB(26);:PRINT USING "$$######.##";TOTAL
        PRINT
        CLOSE 1
6099    RETURN
        REM ***************
7000    REM PRINT BANK BALANCE
        REM ***************
        GOSUB 9000
        IF ANS EQ EXIT THEN 7099
        OPEN "BANK$ACC.DAT" RECL 50 AS 1
        READ # 1,2;BALANCE,LAST.DATE$
        PRINT USING "AS OF  &";DATE$
        PRINT USING "CHECKING ACCOUNT BALANCE $$######.##";BALANCE
        PRINT USING "DATE OF LAST TRANSACTION  &";LAST.DATE$
        PRINT
        READ # 1,3;BALANCE,LAST.DATE$
        PRINT USING "SAVINGS ACCOUNT BALANCE  $$######.##";BALANCE
        PRINT USING "DATE OF LAST TRANSACTION  &";LAST.DATE$
        CLOSE 1
7099    RETURN
        REM ***************
8000    REM EDIT A RECORD
        REM ***************
        INPUT "EDIT PAY STUB OR DEPOSIT RECORD?";RES$
        IF RES$ EQ "EXIT" THEN 8099
        IF LEFT$(RES$,1) NE "P" AND LEFT$(RES$,1) NE "D" THEN \
                GOTO 8000
        IF LEFT$(RES$,1) EQ "D" THEN 8500
        INPUT "PAY STUB CODE?";CODE$
        PRINT
        GOSUB 3500                  REM SEARCH ROUTINE
        IF FAIL EQ YES THEN 8099
        GOSUB 10000       REM READ RECORD
        GOSUB 11500       REM PRINT RECORD
        PRINT
        INPUT "CHANGE WHAT FIELD?";FIELD
        IF FIELD LT STRING.FIELDS+1 THEN \
                PRINT "NEW FIELD";FIELD; :\
                INPUT R$(FIELD) \
        ELSE \
                PRINT "NEW FIELD";FIELD; :\
                INPUT R(FIELD-STRING.FIELDS)
        GOSUB 11000       REM WRITE ADJUSTED RECORD
        CLOSE 1
8099    RETURN
8500    REM ***** EDIT DEPOSIT RECORD *****
        INPUT "AMOUNT AND DATE OF DEPOSIT?";AMOUNT1,DATE.DEP1$
        OPEN "BANK$ACC.DAT" RECL 50 AS 1
        READ # 1,1;RECORDS
```

tem to be changed by a system of fields, identical to the editing scheme of the accounts-payable program. An example of the editing procedure may be found in Sample 4. Samples 1 through 4 also depict the use of the other functions.

```
OPTION (?=LIST OPTIONS)? C
PAY STUB CODE? A518212
PRINT ON TTY? Y

A518212    05/26/78   05/19/77
  1155.07     0.00       0.00   1155.07
     0.00     0.00       0.00   1155.07
  1155.07   239.66    1155.07     69.88      49.81      25.41      11.55
    14.44  1155.07      60.64      0.00       0.00
     0.00     0.00
   683.68

OPTION (?=LIST OPTIONS)? D
STUB CODE? A518212
PAY CHECK VALUE     $683.68
FULL AMOUNT DEPOSITED? N
WHAT AMOUNT TO CHECKING ACCOUNT? 500
WHAT AMOUNT TO SAVINGS ACCOUNT? 183.68
WHAT DATE CHECK DEPOSITED?  *** SPACE ENTERED INSERTS CURRENT DATE ***

OPTION (?=LIST OPTIONS)? E
CHECKING OR SAVINGS ACCOUNT? SAVINGS
AMOUNT OF DEPOSIT? 1000000
MONEY FROM WHOM? KILOBAUD ARTICLE
DATE DEPOSITED?             *** SPACE ENTERED INSERTS CURRENT DATE ***

OPTION (?=LIST OPTIONS)? F
ANY SPECIFIC ACCOUNT? NO
PRINT ON TTY? Y

ACN          ACCOUNT              AMOUNT      DATE DEP
---  ----------------------   ----------  ----------

 C    PAY STUB A518212           $500.00    06/09/78
 S    PAY STUB A518212           $183.68    06/09/78
 S    KILOBAUD ARTICLE       $1000000.00    06/09/78

----------------------------------------------------
                             $1000683.68
```

Sample 3. Functions C, D, E and F.

The Program

The accounts-receivable program, like accounts payable, is written in a structured format. Wherever possible, subroutines have been employed. This has made debugging easier and should do the same for modification. Table 1 provides a list of all blocks of the program and their line numbers, which may be useful if extensive rewriting is being done.

Most of the blocks operate independently of each other, and so they can be modified and even removed while allowing the rest of the program to operate. This is particularly handy if you expect to be making many syntax changes while entering the program. Simply enter the major subroutines first, and after they have been thoroughly tested, begin including the major function blocks. If any problems occur, you will know the error must be in the block most recently entered.

That the major subroutines are shared by all function blocks also allows the package to be easily customized. Suppose, for instance, that two printers are available: a high-speed matrix printer and a slower, letter-quality unit. Only block 9000 need be changed to

```
        FOR RECORD=4 TO RECORDS-1
                GOSUB 10100       REM READ RECORD
                IF AMOUNT EQ AMOUNT1 AND \
                        DATE.DEP$ EQ DATE.DEP1$ THEN 8600
8550    NEXT RECORD
        PRINT "DEPOSIT RECORD NOT FOUND"
        CLOSE 1
        GOTO 8099
8600    PRINT
        PRINT USING F5$;ACC$,ACCOUNT$,AMOUNT,DATE.DEP$
        PRINT
        PRINT "THIS RECORD TO BE EDITED";
        GOSUB 9100                REM GET ANSWER
        IF ANS EQ NO THEN 8550
        INPUT "CHANGE WHAT FIELD?";FIELD
        PRINT "NEW FIELD";FIELD;
        IF FIELD EQ 1 THEN INPUT ACC$
        IF FIELD EQ 2 THEN INPUT ACCOUNT$
        IF FIELD EQ 3 THEN INPUT AMOUNT
        IF FIELD EQ 4 THEN INPUT DATE.DEP$
        GOSUB 11100               REM WRITE CORRECTED RECORD
        CLOSE 1
        GOTO 8099
8999    STOP
        REM ***************
9000    REM TTY QUESTION
        REM ***************
        PRINT "PRINT ON TTY";
        GOSUB 9100                REM GET ANSWER
        IF ANS EQ YES THEN LPRINTER WIDTH 72
        PRINT
        RETURN
        REM ***************
9100    REM YES OR NO ANSWER
        REM ***************
        INPUT RES$
        ANS=2
        IF LEFT$(RES$,1) EQ "Y" THEN ANS=YES
        IF LEFT$(RES$,1) EQ "N" THEN ANS=NO
        IF RES$="EXIT" THEN ANS=EXIT
        IF ANS NE 2 THEN RETURN
        PRINT "PLEASE ANSWER YES OR NO";
        GOTO 9100
        REM ***************
10000   REM READ RECORD FROM DISK
        REM ***************
        REM THIS ROUTINE MUST BE ADAPTED TO YOUR FIELD SCHEME
        READ # 1,RECORD; \
        R$(1),R$(2),R$(3),R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8), \
        R(9),R(10),R(11),R(12),R(13),R(14),R(15),R(16),R(17),R(18), \
        R(19),R(20),R(21),R(22),R(23)
        RETURN
10100   REM ***************
        REM READ DEPOSIT RECORD
        REM ***************
        READ # 1,RECORD;ACC$,ACCOUNT$,AMOUNT,DATE.DEP$
        RETURN
        REM ***************
10500   REM READ RECORD FROM CRT
        REM ***************
        RESTORE
        PRINT "NO DECIMALS PLEASE"
        PRINT
        FOR INDEXA=1 TO STRING.FIELDS
                READ H$
                PRINT H$;
                INPUT R$(INDEXA)
        NEXT INDEXA
        FOR INDEXA=STRING.FIELDS+1 TO NUMERIC.FIELDS+STRING.FIELDS
                READ H$
                PRINT H$;
```

```
OPTION (?=LIST OPTIONS)? G
PRINT ON TTY? MAYBE
PLEASE ANSWER YES OR NO? Y

AS OF  06/09/78
CHECKING ACCOUNT BALANCE       $500.00
DATE OF LAST TRANSACTION   06/09/78

SAVINGS ACCOUNT BALANCE   $1000183.68
DATE OF LAST TRANSACTION   06/09/78

OPTION (?=LIST OPTIONS)? H
EDIT PAY STUB OR DEPOSIT RECORD? PAY STUB
PAY STUB CODE? A518212

A518212    05/26/78   05/19/77
  1155.07       0.00       0.00  1155.07
     0.00       0.00       0.00  1155.07
  1155.07     239.66   1155.07     69.88      49.81       25.41       11.55
    14.44    1155.07     60.64      0.00       0.00
     0.00       0.00
   683.68

CHANGE WHAT FIELD? 3
NEW FIELD 3 ? 05/19/78              *** ERROR CORRECTED ***

OPTION (?=LIST OPTIONS)? I
```

Sample 4. Functions G, H and I.

allow the selection of either printer throughout the program!

Furthermore, a whole new function could be added simply by inserting the new code block where there is space and adding its name and line number to the menu block. Likewise, it is just as easy to delete a function by eliminating its block of code and removing its name and line number from the menu section. (Be careful, however, not to remove an important subroutine! Again, refer to Table 1.)

With regard to adapting the program to run on other than the recommended system, both the accounts-payable and accounts-receivable programs are written in extended ANSI standard BASIC. For this reason, I would advise against using a non-extended BASIC interpreter or compiler, although with some Yankee ingenuity it may be possible. Also, beware of storage problems that may occur when the program is run with an interpreter. The source code requires over 12K, but CBASIC compiles this down to an acceptable 7K (not including space for the compiler and CP/M). Other BASIC interpreters may not be this efficient.

Since the program is highly interactive, disk storage is preferable to tape; otherwise some of the functions will be intolerably time-consuming. Both the memory and the external storage requirements are relatively minor considering the newer and less expensive disk systems and the development of better BASIC systems. Furthermore, memory prices are down (32K for $495—assembled!), so the program's memory needs are within reach. Finally, with the addition of these programs, you will be putting your computer to work, which should make those Star Trek game sessions (in the spare time you have gained by having the computer handle your finances) a lot more enjoyable.

In the third and final installment of the series, I will present a report program that will take the data generated by accounts payable and accounts receivable and formulate charts and tables to be used in preparing tax returns, analyzing spending and income trends and in controlling (we hope) that most elusive of possessions: money. Until then, may your tax returns never be audited! ■

```
              INPUT R(INDEXA-STRING.FIELDS)
        NEXT INDEXA
        PRINT
        FOR INDEXA=1 TO NUMERIC.FIELDS
              R(INDEXA)=R(INDEXA)/100
        NEXT INDEXA
        RETURN
        REM ***************
11000   REM WRITE RECORD TO DISK
        REM ***************
        REM THIS ROUTINE MUST BE ADAPTED TO YOUR FIELD SCHEME
        PRINT # 1,RECORD; \
        R$(1),R$(2),R$(3),R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8), \
        R(9),R(10),R(11),R(12),R(13),R(14),R(15),R(16),R(17),R(18), \
        R(19),R(20),R(21),R(22),R(23)
        RETURN
        REM ***************
11100   REM WRITE DEPOSIT RECORD
        REM ***************
        PRINT # 1,RECORD;ACC$,ACCOUNT$,AMOUNT,DATE.DEP$
        RETURN
        REM ***************
11500   REM WRITE RECORD TO CRT
        REM ***************
        REM THIS ROUTINE SHOULD PRINT YOUR DATA IN THE ORDER IN WHICH
        REM IT APPEARS ON YOUR PAY STUB
        PRINT USING F1$;R$(1),R$(2),R$(3)
        PRINT USING F2$;R(1),R(2),R(3),R(4)
        PRINT USING F2$;R(5),R(6),R(7),R(8)
        PRINT USING F2$;R(9),R(10),R(11),R(12),R(13),R(14),R(15)
        PRINT USING F2$;R(16),R(17),R(18),R(19),R(20)
        PRINT USING F2$;R(21),R(22)
        PRINT USING F2$;R(23)
        RETURN
        END
```

Major routines	
100	Initialization
101	Parameter block
130	Menu print
Function blocks	
1000-1099	Function A: Input new paycheck
2000-2099	Function B: List paychecks
3000-3099	Function C: Print a pay stub
4000-4099	Function D: Deposit an existing paycheck
5000-5099	Function E: Make a miscellaneous deposit
6000-6099	Function F: List deposits
7000-7099	Function G: Print bank balances
8000-8099	Function H: Edit (pay stub)
8500	Function H: Edit (deposit record)
8999	Function I: Return to operating system
Subroutines	
1500-1900	Deposit subroutine
3500-3599	Pay stub search subroutine
9000	Teletype on/off question
9100	Yes/no question
10000	Read pay stub record from disk
10100	Read deposit record from disk
10500	Read pay stub record from console
11000	Write pay stub record to disk
11100	Write deposit record to disk
11500	Write pay stub record to console

Table 1. Program breakdown.

✓ Reader Service—see page 179

Sargon Meets TRS-80

When you confront the two of them, you face formidable opposition.

Ronald H. Bobo
3246 Gravois
St. Louis MO 63118

In the third millennium BC, there was a king of ancient Akkad in the Middle East named Sargon. Some two thousand years later, another king by the same name reigned at Dur Sharrakin in Assyria.

At the West Coast Computer Faire in 1978, a microcomputer program called Sargon won the chess tournament handily.

I don't profess to know which of the ancient gentlemen the 1978 Sargon is named for. What I do profess to know is that it can be a formidable opponent, at least to one of my limited skills.

The Sargon program was written by Dan and Kathe Spracklen of San Diego and adapted for the Radio Shack TRS-80, Level II, 16K machine by Paul Lohnes. Written in Z-80 assembly language, Sargon takes up over 11K. Loading is via the SYSTEM command.

How to Play

There are six levels of difficulty, referred to as "plies," which are half moves. At ply 1, Sargon checks only its own possible moves and makes the one calculated to be best. At ply 2, all of Sargon's moves, as well as all possible responses by the opponent, are checked, and the calculated best move is then made. At ply 6, Sargon is looking ahead for three moves, plus all possible responses.

The computer usually takes 1 or 2 minutes to move when playing at ply 2, but not always. There is a built-in ply adjustment, and computation time increases when things get tough for Sargon. Each higher ply increases the move time and playing ability.

Psyching Out the Opponent

In my first game with Sargon I started out at ply 2, playing white, and used the same opening with which I toppled another popular micro chess program in seven moves. It didn't work. Within 20 moves, my position was so hopeless I resigned, thoroughly humiliated.

Next game, the ply was set to 3 and I played white again. Sargon responded almost instantaneously to my first move. I moved again and waited . . . and waited . . . and waited some more. Finally, the second move came.

Intrigued, I began to keep notes on the computer's response time per move. In the early part of the game, it averaged 13 to 15 minutes. By midgame, 30 minutes or slightly over was par. We didn't get to the end of the game before bedtime, 5 1/2 hours into the game.

Next day was Sunday. After losing another game at ply 2, I decided to investigate ply 6. After all, if a move takes half an hour at ply 3, then ply 6 should take twice as long, right? Wrong. The game started at 3 PM. Again, Sargon's response to my first move was made swiftly. I moved again, then settled back to do some reading, keeping one eye on the CRT so I could note the exact time of the move.

Seven and one-half hours later, Sargon still hadn't moved! I wanted to use the computer for something else, so I terminated the game.

Later, in a phone interview with Kathe Spracklen, I asked how long a move might be expected to take at ply 6. She chuckled and said, "Days."

Mrs. Spracklen suggested playing at ply 2 until near game's end when fewer pieces are on the board, then adjusting the ply upward, since the end-game algorithm is not as strong as earlier portions of the program. With fewer pieces, of course, move time would be considerably reduced.

The Game

Although the TRS-80 graphics for Sargon are well done, the pieces take getting used to. The board squares are solid black and white, and as a result, when a piece lands on a square of the same color, it has to change its own color so that it may still be seen. You must then check the center of the piece, where a small area of its true color will be discernible. This is not as inconvenient as it may sound, and, after a game or two, you'll have it down pat.

Moves are made using algebraic notation—type in the square that the piece currently occupies, a hyphen and the square to which you are moving (i.e., E2-E4 would move the white king's pawn two spaces forward). *En passant* pawn captures are provided for, as well as castling.

The back-space key, by the way, is disabled. If you hit the wrong key by mistake, or a key repeats (somewhat common with the TRS-80), you'll get an error message and have to start the move over.

A record of both the player's and Sargon's moves is listed along the left side of the board in algebraic notation. When the bottom of the screen is reached, the listing is cleared and the display starts over from the current move.

When playing white, Sargon is programmed to alternate randomly between opening with the king's and queen's pawns, thus making for a little variety. Playing black, it seems to al-

ways respond identically to identical situtations.

In the event you make a mistake and want to take it back, a procedure exists for doing just that. It may seem a bit complicated at first, but follow the instruction pamphlet carefully to avoid any problems. This same routine can also be used to set up the board for solving specific chess problems.

There are a few additions I would like to see on future versions.

1. A display of the algebraic designations for each square, callable on command, would be helpful to those not familiar with this notation.

2. A reversal of the board when playing black. As it is now, white remains at the bottom of the screen at all times so, when playing black, one must, in effect, look at the board upside down. Because of this, I play only white.

3. A display to indicate that Sargon is calculating its move. When playing at any level above ply 2, I wonder whether the program is operating, or if, perhaps, the computer has broken down. There is really no way to tell, until Sargon makes its move.

The Sargon program cassette is published by the Hayden Book Company, Inc., of Rochelle Park, New Jersey, and is available at many computer stores.

For the do-it-yourselfer, a book entitled *SARGON: A Computer Chess Program* is available from the same publisher. The book contains a block diagram of the program along with a Z-80 assembly-language listing and an index to the 50 subroutines. Price is $14.95 for the book; $19.95 for a TRS-80 cassette. Sargon cassettes are also available for the PET and Apple II machines.

I must rate this chess program an excellent buy for anyone who loves the game but doesn't always have a human opponent available.

Oh, yes . . . in six weeks, I have only won three times. ■

Clement S. Pepper
3270-96 Caminito East Bluff
La Jolla CA 92037

Safe Ports

Protect your I/O ports with this bidirectional buffer.

The individually addressable I/O ports of my KIM-1 are a valuable feature I really prize. I use them a lot for a variety of circuit experiments. I have also been haunted by the fear that through some program or circuitry defect I would inadvertently write a "0" to a low-impedance 5 volt source. Replacing an expensive 40-pin DIP is not a task I anticipate with enthusiasm.

So I have been scanning the literature in an unsuccessful search for a bidirectional buffer IC. My objective is a transparent circuit I can connect and forget.

Since I didn't find what I was looking for ready-made, I devised a circuit of my own. In addition to safety, it has another benefit. The drive capability is several TTL loads — from five to ten depending on the combined tolerances of the devices.

"Why not," I mused over a cup of coffee, "connect two comparators so that the output of each is summed into the input of the other! Then signals will flow both ways . . . and presto! I'll have a bidirectional gate." It took a little bit more than that, but not a whole lot more. The result is displayed in the circuit of Fig. 1.

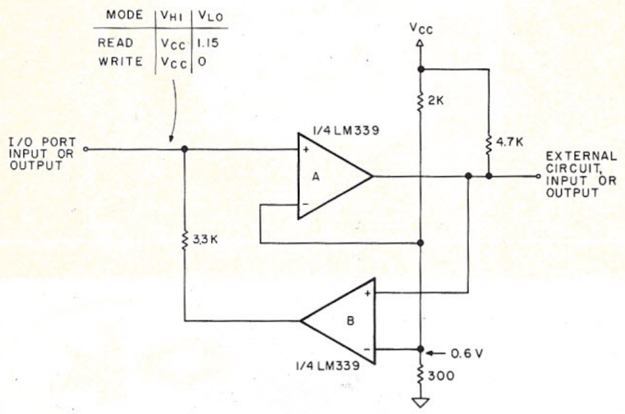


Fig. 1. The bidirectional buffer. When connected to the KIM-1 I/O port it will both read and write to an external circuit. The buffer will also sink or source a greater current than the port.

How It Works

The circuit is based on the capabilities of the LM339 comparator. This versatile device will operate from a single power supply with a range of +2 to +36 volts. The common mode input range includes ground. Input impedance is high; the typical bias current is 25 nA. The typical input offset is 3 mV. Best of all, the output is the open collector of an uncommitted transistor. A schematic of the internal circuitry and package pinning is shown in Fig. 2.

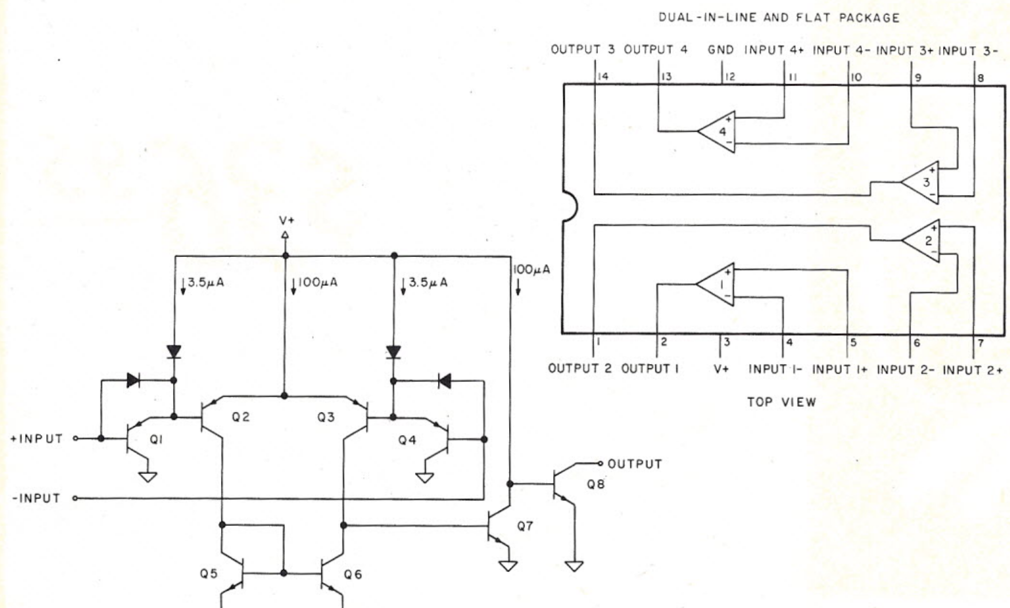


Fig. 2. LM339 internal schematic and pin connection diagrams.

Now let's look at KIM's I/O ports. My milliammeter tells me a port delivers 0.4 mA from a nearly constant current source in both the read and write mode. That is, the same current flows into ground when the port is shorted as flows into the 3300 Ohm resistor shown in the circuit.

With the port open, the output equals V_{CC}. The switching point is the center of V_{CC}, about 2.5 volts. The 6530 data sheet specs the output high current sourcing as 100 uA minimum, with 1 mA typical.

It is the READ current that dictates the design, provided that an equal or greater current is sourced in the WRITE mode. If the current is too low, the gate input will not overcome the reference. If it is too high, such that it stays above 2.5 volts, the port will not read a zero. So it will be wise to measure the read current from your ports and revise the input resistor if necessary to maintain the 1.1 volt potential. There is only one current sinking mode; when writing a "0" the port terminal goes low.

So let's see how the circuit does its thing. Observe the table in the upper left of Fig. 1. Port voltages in the high and low states are given for the READ and WRITE modes.

Consider the READ first. VHI or VLO refers to the external cir-

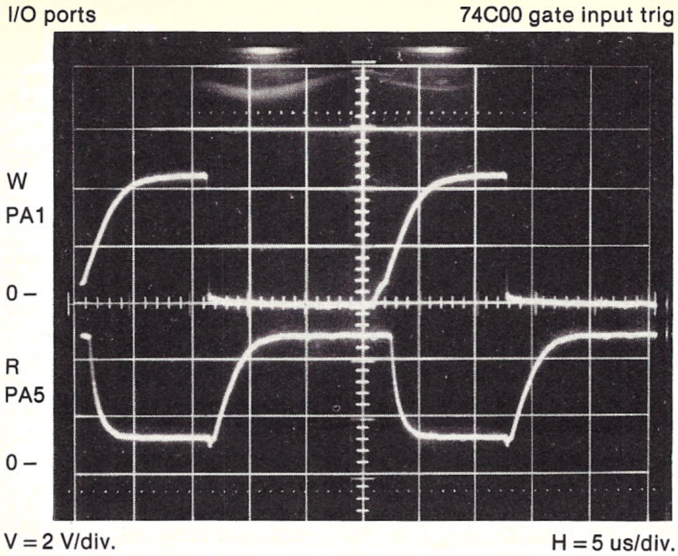


Photo 1. The upper trace is the I/O port write instructions. The lower trace is the port read signal. Observe that the trace bottoms out at about 1.2 volts when reading a zero. This is well below the switching threshold for the port.

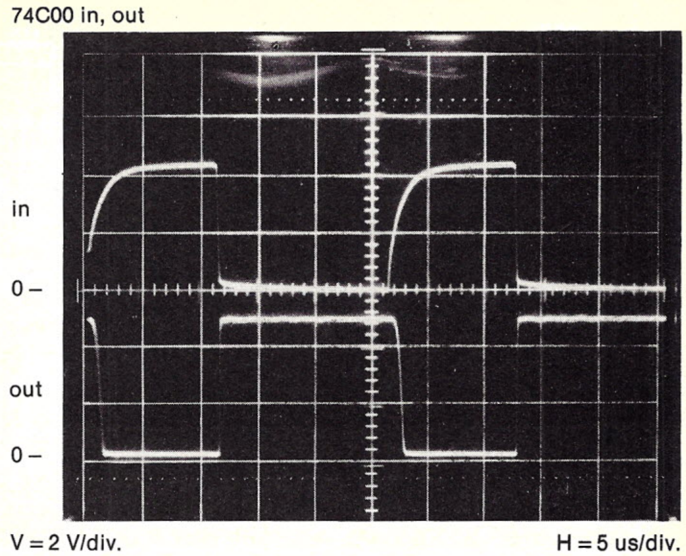


Photo 2. The upper trace is the NAND input. The lower trace is the NAND output. The gate is a 74C00.

cuit. Observe that the non-inverting inputs of both comparators are connected to a reference of 0.6 volts. This establishes the switching threshold for the comparators.

An input from the external circuit exceeding the threshold will cause the "B" output to go high. Since the output is a transistor in cutoff, no current flows through the 3.3k resistor. The input of "A" draws negligible current; the port output is at V_{CC}. The "A" output is also cut off and has no effect on the external circuit.

Now suppose the external input goes to zero. The output of "B" now goes low, and the port sources current through the 3.3k load. This pulls the port potential down to 1.1 volts. Since this is well below 2.5 volts, the port reads a "0." However, the "A" input still exceeds the threshold, and the output remains in cutoff. This is essential, for should it go low, the gate will be locked up.

Now let's write a "1" to our external circuit. We will also assume a worst-case situation in which the external circuit is in the zero state. The output of "B" is low, drawing current from the port. Still, 1.1 V > 0.6 V and the output of "A" is off. Current flow through the 4.7k resistor pulls up the input to the external circuit and a "1" is written.

Next we elect to write a "0." The port goes low and pulls down the input of "A" below the threshold. Presto. "A" turns on, its output goes low and the zero is written. The zero is read back by "B," its output also goes low, but this has no effect.

Fig. 3 provides an illustration of a working circuit. The buffer is writing into a NAND gate from one port and reading back with a second. I wrote a looping routine for this and viewed the operation on my oscilloscope. Scope traces of the circuit in Fig. 3 are provided in Photos 1 and 2.

The upper trace of Photo 1 is the I/O write pulse at the KIM port. The lower trace is the I/O read. You can see that the bottom of the read is about 1.2 V above ground (vertical scale is 2 V per division). A 74C00 CMOS NAND is the external device. The 74C00 input and output are shown in the upper and lower traces of Photo 2.

Construction

I built up the circuit on perfboard. The layout is shown in Fig. 4. Eight LM339s are required for the 15 ports. Fourteen of the circuits are duplicates of Fig. 3 (don't put in the NAND!). Port PB7 requires a pull-up. Connect a 10k resistor between the port terminal and the +5 V bus.

An AP Products 217L solderless plug-in matrix was split down the middle and one-half placed at each end as shown for input and output connections. Homemade ribbon cables provide quick connection to KIM at the input and to my external circuits at the output.

Only a single reference line is required for all the comparators. That is, a single 2k and 300 Ohm resistor are wired as shown, and pins 4, 6, 8 and 10 for all the devices are bused together.

Performance Testing

I wasn't about to connect the assembled circuit to my KIM-1 without a bench test. I did not solder the comparators to the board . . . I used sockets. So before plugging in the ICs, I connected the 5 volt power to the bus and checked each socket pin for the correct voltages — 5, 0.6 or 0.

I then disconnected the power and plugged in the comparators. I gave each gate a performance test; for this I simulated the port with a test lead through a 10k pull-up to 5 volts. A voltmeter monitored the port

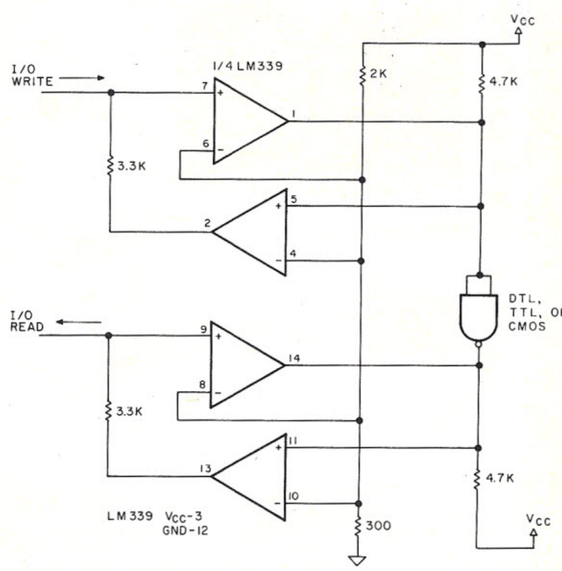


Fig. 3. An illustrative example of the buffer in operation. The upper port provides input to the NAND gate. The lower port reads the gate output.

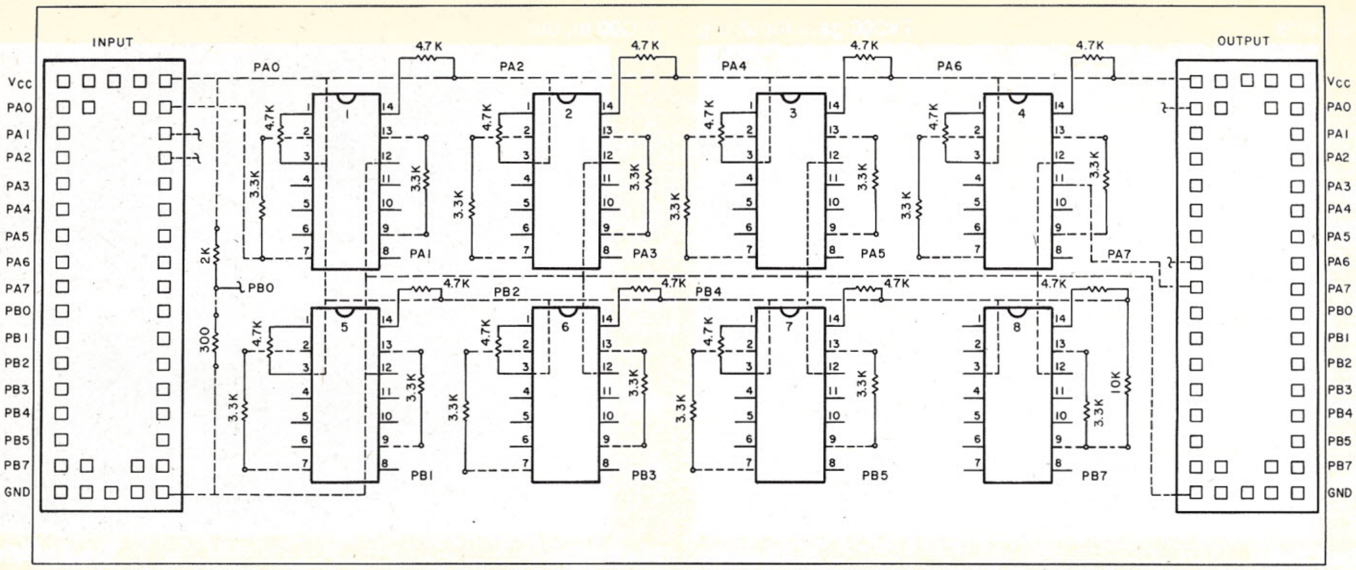


Fig. 4. Perfboard construction is efficient and compact. Circuit Stik Quik Circuits copper stick-ons enhance IC mounting. Upper row buffers ports PA0-PA7; lower row, PB0-PB7. Homemade ribbon cable provides quick connection to KIM and external circuits.

connection to the gate. With the gate output open, the meter will read 5 volts. Grounding the output will pull the reading down to about 1.2 volts. This test paid off — I found a bad solder joint. A higher reading will appear on PB7 because of the pull-up. It will be about 2.1 volts.

The KIM-1 Test

I connected my new gate assembly to KIM's dc power and the two ports with the power off. I then powered and reset KIM. Port A (1700) displayed FF; port B (1702) displayed BF as it should, since PB6 is always zero.

Next I grounded each gate output in turn, observing the port readout for the correct value. All went well.

That concluded the reading portion of the test. Now, will it write? I keyed in a short read, write and jump back to read routine and stepped it through. Bravo!

In my use of the gate so far, with both TTL and CMOS devices, it appears to be doing exactly what it was designed to do. Using the ports without fear is a good feeling. ■

✓ Reader Service—see page 179

Projecting Future Profits

This business-simulation program will help you get started in business programming.

Ernie Brooner
Box 236
Lakeside MT 59922

This simple program demonstrates a useful but unusual small-business computer application. It is a good exercise for the programmer just starting in business software, since it is short and obtains its input from the keyboard, thus not requiring disk files or a large memory.

The Break-Even Point

The concept of the "break-even point" is widely taught in business schools. It takes into account that each business has a minimum of expenses, such as rent, even if it sells nothing. Variable costs, on the other hand, increase in direct proportion to sales or production. This is usually expressed using a linear graph such as that shown in Fig. 1, where a horizontal line represents the fixed costs and a slope stands for the variable costs. The total costs are represented by a sloping line, which is the sum of the other two. Sales is also represented by a sloping line, and its range is zero to maximum on the graph.

At some point the sales line and total-cost lines intersect. This is the break-even point. Losses are below this line; profits are above it. Notice that the percentage of gross profit constantly changes with volume. This concept is used, for example, to determine whether or not it will be worthwhile to make certain changes in the business, such as hiring additional clerks or changing the hours of operation.

This problem can be a lot of fun on a graphics terminal; however, the program is written without graphics and the results are output to a printer, which simplifies things and permits a wider variety of hardware to be used. The listing is in North Star BASIC and can easily be translated to use with other interpreters. In many cases it will be necessary to replace the North Star format statement (i.e., %12F2 as used in line 170) with IMAGE and PRINT USING instructions, or else just omit the fancy columnizing.

Operation

In its opening, the program prompts the user to enter the pertinent information, that is, the fixed costs, the maximum volume possible with the existing facilities and the variable costs that would be incurred at maximum sales. The program then calculates the break-even point and prints it in dollars and as a percentage. Next, it prints a table of figures showing—for each 10 percent increment—the sales, costs and profit. It then prompts the operator to enter new data and try again.

Several possible scenarios may be compared side by side for planning purposes. There is a small amount of rounding since percentage is considered in increments of one; the break-even point is the next highest one-percent increment. ■

```
10  REM INPUT DATA USED TO CALCULATE BREAK-EVEN POINT
20  INPUT "FIXED COST OF OPERATION                ",F
30  INPUT "MAX SALES IN $ AT FULL PRODUCTION    ",M
40  INPUT "VARIABLE COST FOR FULL PRODUCTION    ",V
50  PRINT
60  PRINT "FIXED COSTS = ", %12F2,F
70  PRINT "MAX SALES  = ",%12F2,M
80  PRINT "VARIABLE COSTS = ",%9F2,V
90  PRINT
100 REM CALCULATIONS
110 FOR X = 1 TO 100
120   S = X*(M/100)
130   E = (X*(V/100)) + F
140   IF E<S THEN 170
150 NEXT
160 REM PRINT THE RESULTS
170 PRINT "BREAK-EVEN POINT IS",TAB(25),%12F2,S,"  ",
180 PRINT X,"    PERCENT"
190 PRINT
200 M1 = M/10
210 V1 = V/10
220 M2 = M1
230 V2 = V1
240 PRINT TAB(7), "SALES",TAB(19), "COSTS",TAB(30),"PROFIT"
250 FOR X = 1 TO 10
260   PRINT    %12F2,INT(M2),(((V2/M2)*M2) + F),
270   PRINT TAB(25),%12F2, INT(M2) - (((V2/M2)*M2) + F)
280   REM INCREMENT VALUES
290   M2 = M2 + M1
300   V2 = V2 + V1
310   NEXT
320 PRINT
330 GOTO 20
FIXED COST OF OPERATION                      10000
MAX SALES IN $ AT FULL PRODUCTION            100000
VARIABLE COST FOR FULL PRODUCTION             66000
FIXED COSTS =        10000.00
MAX SALES =         100000.00
VARIABLE COSTS =     66000.00
BREAK-EVEN POINT IS                  30000.00   30   PERCENT
  SALES        COSTS       PROFIT
  10000.00     16600.00    -6600.00
  20000.00     23200.00    -3200.00
  30000.00     29800.00      200.00
  40000.00     36400.00     3600.00
  50000.00     43000.00     7000.00
  60000.00     49600.00    10400.00
  70000.00     56200.00    13800.00
  80000.00     62800.00    17200.00
  90000.00     69400.00    20600.00
 100000.00     76000.00    24000.00
```

Program listing.

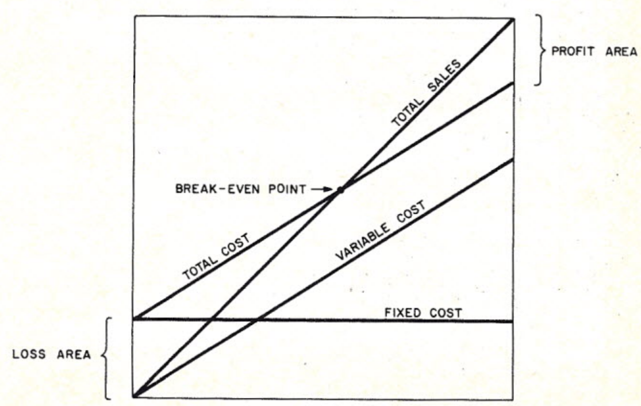


Fig. 1. Break-even chart (not to scale).

Randomness Is More Than It Seems

As this article points out, the term "random number" can be a misnomer.

John R. Cameron
Alette M. Comrack
PO Box 1517
Palo Alto CA 94301

Random numbers are an important part of many game programs and are often used in sampling procedures. Most higher languages include a random-number-generating function, which you usually use without a second thought. However, do you *really* know whether or not this deceptively simple function returns a random set of numbers?

Actually, analyzing the randomness of a set of numbers is a difficult mathematical process and is only rarely complete, since unequal overall distribution and all fixed relationships between numbers in the set must be excluded. Furthermore, relying on mere observation to determine the absence of patterns in a set of random numbers might miss nonrandomness, which relatively simple statistical methods could disclose.

This article presents a BASIC program that deals with these problems and can be used to expose much of the nonrandomness that would likely be encountered in a set of numbers obtained using a random-number generator. It is an extension of the article "Randomness Is Wonderful" by Bill Rogers (October 1978 *Kilobaud*, p. 62). That article dealt with the problem of nonrandomness and presented a BASIC program that displayed the overall distribution of a set of random numbers. As we shall see, Rogers' technique is limited since it neither exposes all cases of unequal distribution nor treats the problem of repeatability, which was addressed in the article's introduction.

```
10 REM Production of Random Number Set
20 RANDOMIZE:DIM RN(2000):PRECISION 2
30 FOR M=1 TO 2000
40 RN(M)=RND(1)
50 IF RN(M)=1 GOTO 40
60 NEXT M
70 REM Evaluation of Frequency Distribution
80 C2=0:FOR X=1 TO 10:N(X)=0:NEXT X
90 FOR M=1 TO 2000
100 X=INT(RN(M)*10)+1
110 N(X)=N(X)+1:NEXT M
120 ? " X    1    2    3    4    5    6    7    8    9   10"
    :? " %   ";
130 FOR X=1 TO 10
140 IF N(X)<200 THEN ? " ";
150 ? INT(N(X)/20);
160 C2=C2+(N(X)-200)^2/200:NEXT X
170 ?:?:? "Chi-Squared Value Is";C2;"for df=9":?
180 C2=0' Evaluation of Pairwise Distribution
190 FOR Y=1 TO 10
200 FOR X=1 TO 10:N2(Y,X)=0:NEXT X
210 NEXT Y
220 FOR M=1 TO 1999 STEP 2
230 Y=INT(RN(M)*10)+1:X=INT(RN(M+1)*10)+1
240 N2(Y,X)=N2(Y,X)+1:NEXT M
250 ? " \X   1    2    3    4    5    6    7    8    9   10"
    :? " Y"
260 FOR Y=1 TO 10:? Y;:IF Y<10 THEN ? " ";
270 FOR X=1 TO 10
280 IF N2(Y,X)<10 THEN ? " ";
290 ? N2(Y,X);
300 C2=C2+(N2(Y,X)-10)^2/10:NEXT X
310 ?:NEXT Y
320 ?:? "Chi-Squared Value Is";C2;"for df=99":?
330 END
```

Program listing.

The Program

First, we will describe the program and then discuss some examples. The program is written in Xitan SuperBASIC and is divided into three potentially independent parts. The first part (lines 10-60) generates 2000 random numbers (values between 0 and 1) using RND(1) and stores them in a one-dimensional array denoted by RN(M). It excludes for later computational purposes any random number equal to one. All calculations in this program are made to 12 significant places, but only two are generally printed (?) according to the PRECISION 2 command.

If you need to conserve memory, you can alter the program to put each random-number-generation iteration just before the evaluation step in the latter parts of the program. We stored the numbers in an array to facilitate the application of different evaluation techniques to the same set of numbers. For the same reason, both the second and third parts of the program start by clearing their internal variables (counters).

The second part (lines 70-170) examines the overall distribution of the RN(M) set by counting the number, N(X), of random numbers whose values fall into each of ten equal-size slots labeled 1 to 10 (i.e., slot 1, N(1), is the count of numbers whose value is ≥0 and <0.1).

This program, like Rogers', examines the frequency of each possible first digit in a set of random numbers. It could easily be modified to examine the other, often overlooked, digits. On the average, you can expect 200 numbers in each slot for a frequency of 10 percent (printed as the integral percent).

An added feature is the calculation of a statistical value, chi-square. This value is the sum of the ten N(X) slot counters minus the expected value (200), each squared and then divided by the expected value. Chi-square and the value of df (degrees of freedom) are used to determine the probability that the distribution is random.

Chi-square obviously increases as the various numbers obtained differ more from the expected value (200); df simply indicates the number of data values used in the calculation of chi-square, which can vary independently. In this case, df = 9 since the tenth slot number must always equal 2000 minus the sum of the other nine; it is,

Percent Probabilities							
99	95	90	50	10	5	1	0.1
Chi^2 for df=9							
2.1	3.3	4.2	8.3	15	17	22	28
Chi^2 for df=99							
69	77	81	99	117	123	134	147

***Table 1.** Probabilities in percent that the chi-square of a random set with the same degrees of freedom (df) will have a value greater than the one given.*

therefore, not independent.

The third part of the program (lines 180-320) is similar to the second part except that it examines the set of random numbers as 1000 pairs to identify whether the value of the first digit of a number affects the value of the first digit of the next consecutive number in the set. Note that this approach supplants the other technique in that it examines overall distribution as well as pair relationships. It would *not*, however, reliably expose a pattern involving relationships between nonconsecutive numbers.

Each of the 1000 consecutive pairs of random numbers in the set is evaluated to determine into which of the 100 slots it falls. The slots are arranged in a 10 by 10 matrix, N2(Y,X), according to the value of the first digit, as before. The counter value for this slot is then incremented by one.

The y-axis of 1 to 10 denotes the first number in the set, and the x-axis similarly denotes the second. The average expected value in each slot is 10, and the chi-square is calculated as before. This time the value of df is 99 with one slot's value dependent upon the 99 others.

Application

The application of the program involves simply running it (it will take a minute or two) and then examining the values of chi-square obtained. Table 1 gives the probabilities associated with different chi-square values for df = 9 and df = 99. The value given is the probability in percent that the chi-square obtained is smaller than that expected from a random set of data.

Your values should hover around 50 percent. A probability of less than 10 percent indicates nonrandomness (especially if obtained repeatedly), and a value repeatedly greater than 90 percent indicates that the set is more evenly distributed than would be expected. The latter would occur, for instance, if the set consisted of equal frequencies of the ten digits (chi-square would be 0).

Three examples of the program's output are shown to illustrate a few final points. Example 1 is a straight test of our BASIC random-number generator and gives typical results near 50 percent probability in each case (chi-squares of 9.6, df = 9 and 94, df = 99). Ten runs gave average chi-squares of 8.8 and 96 (with standard deviations of 2.3 and 4, respectively, for the statistics-minded reader). Casual examination of the printed values in the second array indicates that they vary greatly from 10. However, the variation is what is expected statistically.

Example 2 shows the result of using RANDOMIZE before every twentieth use of the RND (1) function in line 40 instead of just once in line 20. Note that rather large inequalities of distribution occur in both tests, with probabilities of 1 percent and much less than 0.1 percent that the set of "random" numbers is actually random. Apparently, overuse of RANDOMIZE with Xitan's SuperBASIC is detrimental.

This result is unlike that obtained using Rogers' unnamed BASIC interpreter. Use of RANDOMIZE before each iteration of line 40 (not shown) results in chi-squares of several thousand due to most of the "random" numbers' falling into a few slots in each array.

Observation of the integral frequency percentages shown in the first array of Example 2 does not suggest great disparity from the expected value of ten. However, there is only one chance in 100 that this distribution is actually random. This illustrates the advantage of using a quantitative approach to data analysis. Chi-square tests are often used to demonstrate the significance of differences between the expected and observed frequencies of events in other applications; their probability values for different degrees of freedom can be found in statistical handbooks.

Example 3 demonstrates the advantage of examining the distribution of pairs. An alteration was made in the random-number-generating portion of the program, which made each twentieth random number in the set equal to the previous one (5 percent repetition). Such an alteration does not significantly affect the overall distribution of single values as shown by the chi-square value of 11 for the first array. However, the chi-square value for the second array is 190, giving a probability of much less than 0.1 percent that the array is random.

In fact, it is possible to detect the expected increase in values of the array on the diagonal from upper left to lower right, where X=Y. Some dependency of a random number on the one obtained previously would be a likely result of nonrandomness in a BASIC randomizing operation that carries a seed through the same series of manipulations for each random number generated. For this reason, numbers generated by this method are properly called "pseudorandom," since they are related by whatever operations are used in their generation.

We hope we've convinced you that such patterns of relatedness between consecutive numbers may be observed through the examination of pair distributions but cannot be found by examining only the distribution of single random numbers. ■

X	1	2	3	4	5	6	7	8	9	10
%	11	8	9	10	10	10	9	9	9	9

Chi-Squared Value Is 9.6 for df=9

\X Y	1	2	3	4	5	6	7	8	9	10
1	15	8	9	19	10	4	14	10	16	11
2	10	6	13	11	9	13	6	6	8	14
3	11	5	9	8	10	12	14	11	9	7
4	7	5	7	15	14	6	8	12	10	9
5	11	12	8	9	12	17	8	12	11	11
6	9	11	10	12	10	15	10	8	7	17
7	6	10	8	9	9	7	13	12	10	10
8	17	3	12	10	14	10	9	6	6	10
9	9	9	9	11	11	14	10	13	8	6
10	10	11	11	8	8	9	3	10	7	11

Chi-Squared Value Is 94 for df=99

Example 1.

X	1	2	3	4	5	6	7	8	9	10
%	11	10	7	10	8	9	10	9	10	11

Chi-Squared Value Is 23 for df=9

\X Y	1	2	3	4	5	6	7	8	9	10
1	14	18	9	9	7	9	12	10	6	16
2	10	9	8	10	6	18	17	11	12	8
3	9	7	7	7	11	11	10	3	6	8
4	14	10	6	14	9	11	8	18	10	13
5	8	21	11	3	8	4	14	10	9	11
6	17	1	9	11	3	5	11	18	10	11
7	16	1	8	17	3	11	8	12	18	6
8	4	7	6	13	3	14	4	6	7	20
9	9	8	5	4	15	13	17	8	12	19
10	13	9	10	11	5	7	18	6	11	10

Chi-Squared Value Is 2.0E+02 for df=99

Example 2.

X	1	2	3	4	5	6	7	8	9	10
%	9	9	11	9	9	10	9	9	11	9

Chi-Squared Value Is 11 for df=9

\X Y	1	2	3	4	5	6	7	8	9	10
1	17	8	11	11	7	12	4	7	9	9
2	8	22	6	9	14	6	6	4	10	9
3	9	7	20	11	12	9	11	16	8	9
4	7	7	11	25	7	10	5	5	7	6
5	15	6	14	7	17	10	9	5	9	8
6	10	10	8	11	9	16	11	17	12	11
7	8	7	7	11	8	12	18	7	11	7
8	10	8	9	4	7	12	10	21	10	6
9	4	6	10	9	11	12	7	8	27	14
10	12	5	14	11	6	4	5	12	9	15

Chi-Squared Value Is 1.9E+02 for df=99

Example 3.

OSI's Superboard II

The author saw the ads, bought the board and wrote the following review.

Bruce S. Chamberlain
1982 W. Bayshore Road, #42
Palo Alto CA 94303

Most of you have probably seen the full-page ads for the Ohio Scientific Superboard II, but before buying one you would like to know more about it. So I decided to review the Superboard II single-board computer. At $279 for a board with CPU, 4K RAM, regular 53-key alphanumeric keyboard, Kansas City Standard cassette tape interface, video interface, 2K monitor and 8K Microsoft BASIC in ROM, it has to be the biggest bargain around. All you need to make it operational is a 5 volt 3 Amp power supply and a video monitor or a television with a radio-frequency converter. If you add a cassette recorder to the system for program and data storage, you will be in business.

I live in Palo Alto, California, the heart of Silicon Valley. There are numerous computer stores around, but none of them carry Ohio Scientific products. While down in Los Angeles recently, I checked the computer stores there. The first store, a Byte Shop, told me that the nearest Ohio Scientific distributor was located about a mile away in Huntington Beach.

At the distributor's I looked at the board and read a pamphlet called "The Challenger IP Technical Report." I was so impressed by what Ohio Scientific claimed for their Superboard II that I ordered one. You can have a complete system for $529, which includes the Superboard II and a power supply in a cabinet, a television modified for direct video and a cassette recorder. This is the equivalent of computer systems costing $800 to $1000.

Hardware

When I ordered the Superboard II, the salesman said I would receive it in about three weeks. Much to my surprise, it arrived in two and a half weeks. The single printed circuit board with keyboard was packed in a box of Styrofoam and wrapped in aluminum foil. Packaged this way the board will survive the roughest handling unscathed. The box also contained cassette and video cables, four pamphlets, referenced at the end of this article, and a cassette with six programs: a math tutor for addition, subtraction, multiplication and division; a trigonometry tutor; a teaching program for children on counting from one to ten; a program for balancing your checkbook; a video game called "Star Wars"; and 20 historical questions about various presidents.

Looking over the 12 by 14 1/2 inch printed circuit board, I was surprised to see that it had been hand-soldered. This board has no solder mask or silk-screened legends. The board contains one 40-pin IC, eight 24-pin ICs, ten 18-pin ICs, twenty-seven 16- and 14-pin ICs, one 8-pin IC and ten empty sockets (for eight RAM ICs and two buffer ICs) for a total of 57 ICs. A few resistors, capacitors, a crystal and the keyboard completed the package. The board also has a fuse and reverse protection diode, so if you plug the 5 volt power supply in backwards (shudder), the fuse, rather than all the ICs, blows. The parts for the RS-232 interface and power supply are not provided with the board. In addition there are pads for stuffing four 16-pin ICs for prototyping.

The keyboard is a 53-key ASCII-type keyboard. It features a scanned, rather than decoded, array, so keys can be programmed for user functions. With the shift-lock key down (the shift-lock key is mislabeled, it actually puts the keyboard in a different mode rather than locking it in the shift mode), the keyboard generates uppercase and numerals. Holding either shift key down will generate punctuation marks.

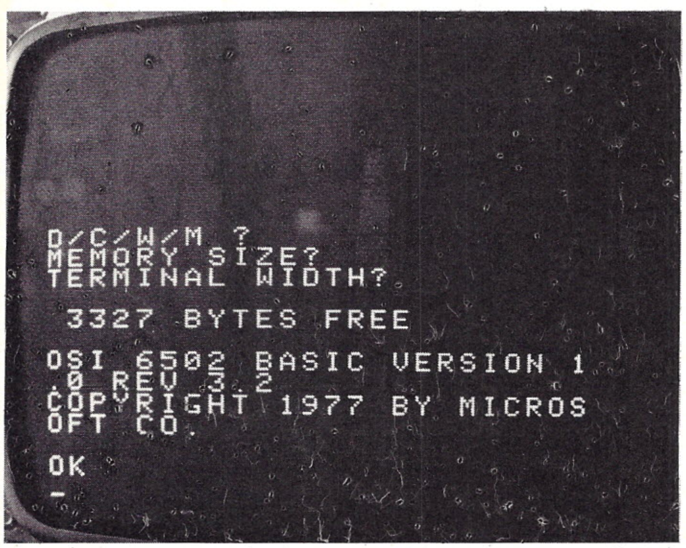


Photo 1. Display after the start-up procedure.

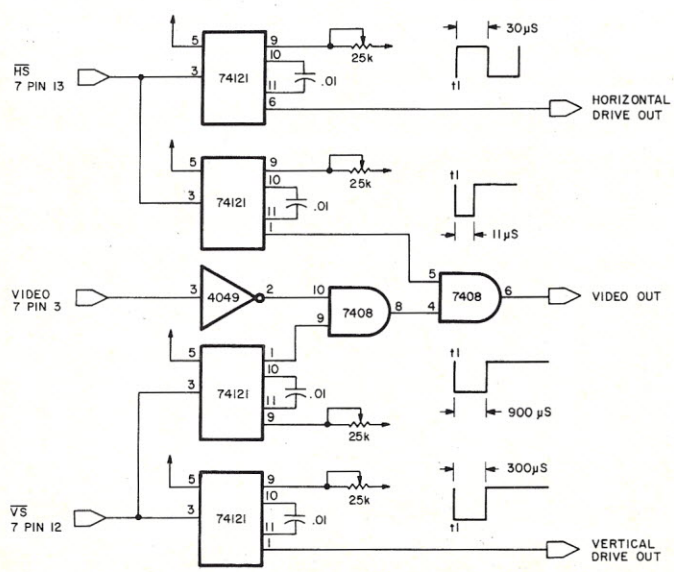


Fig. 1. The interface circuit.

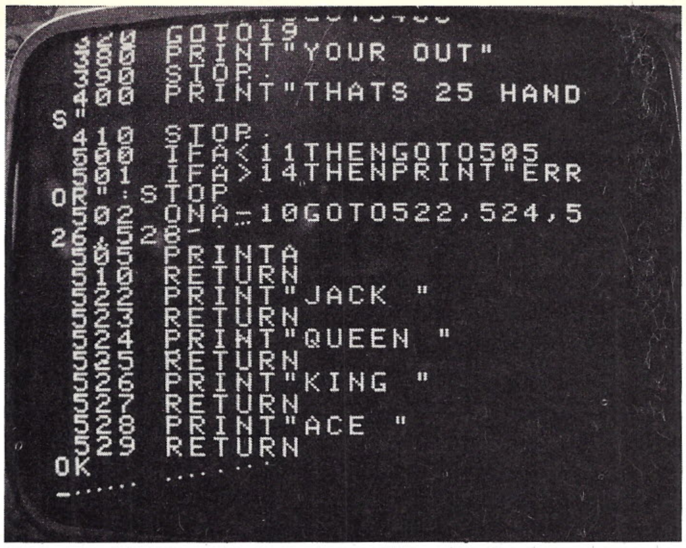


Photo 2. Diagonal bright spots affect the display.

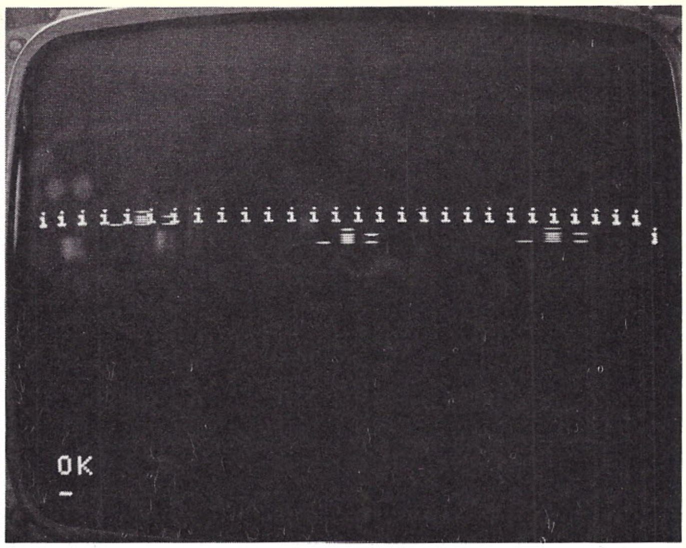


Photo 3. 27 characters per line with an additional four characters on retrace.

The shift-lock key must be in this position for programming and normal system operations.

With the shift-lock key up, the keyboard generates lowercase characters. However, the right and left shift keys act differently. Holding down the left shift key generates uppercase characters and numerals. Holding down the right shift key generates uppercase punctuation. Except when you want lowercase, keep the shift-lock key down. There is a 12-pin Molex connector beside the keyboard for hooking other keys, joysticks, etc., across the keyboard switches for games and other control functions.

The microprocessor is the popular 6502, which is used in KIM-1, Apple II and PET. The board has ten 2114s with sockets for eight more RAM ICs, which are available for $69 from Ohio Scientific. Eight 2114s are used as 4K user memory, giving enough memory to get started writing programs in BASIC. The other two 2114s form 1K of video memory, the contents of which are shown on the television screen.

Input and output consists of a composite video interface that will give you 30 characters by 30 lines on a television without overscan; you will only get 24 by 24 with an ordinary television set because of overscan. Overscan means the beam starts off screen and finishes off screen, both horizontally and vertically.

The other interface is a Kansas City Standard cassette interface, which operates at 300 baud. Ohio Scientific supplies two cables that plug into the back of the board: One plugs into the microphone jack and the other into the earphone jack of a cassette recorder.

To load a tape you type LOAD, start the tape and push RETURN. As it is loading, the screen displays each line of the BASIC program. This way you see what is actually being loaded, and you know immediately if it is loading incorrectly. To save a program, you type SAVE, push RETURN, type LIST, start the tape and push RETURN, causing the tape to store the program as listed, line by line. What could be easier? In all the tapes I loaded and unloaded, I did not have one error and I was using an inexpensive cassette player.

The input/output lines are all through two 12-pin Molex connectors across the back of the board. There is no parallel port. There is a 40-pin socket near the microprocessor IC for expansion. Ohio Scientific sells an expansion board 610 that plugs into this socket. The board mounts over the main board and contains 8K RAM with sockets for another 16K of RAM, a mini-floppy disk interface that will control two mini disk drivers, a real-time clock and expansion interface for a

Photo 4. The interface circuit solved the problem of the diagonal bright spots.

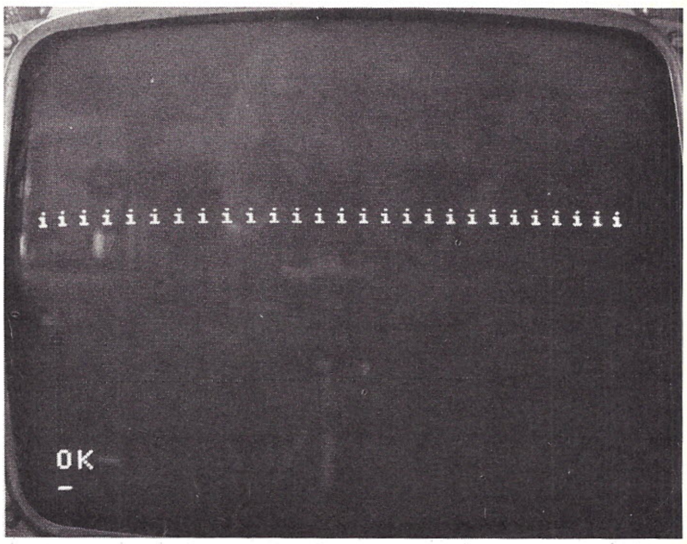


Photo 5. The interface circuit eliminated the retrace characters.

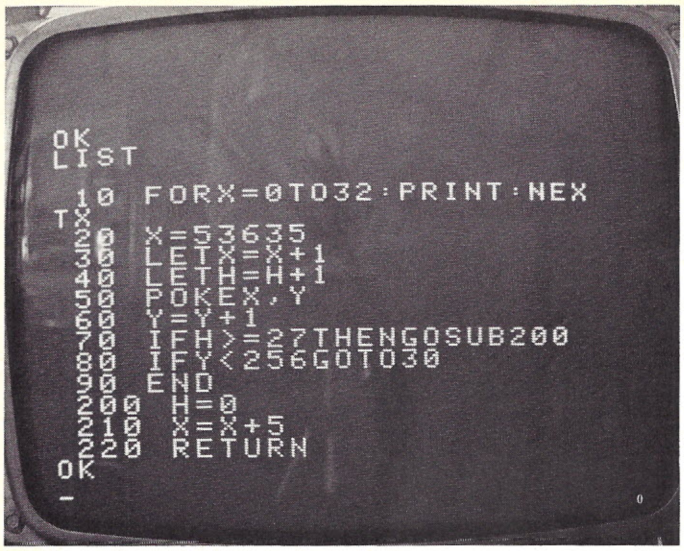


Photo 6. Graphics characters program.

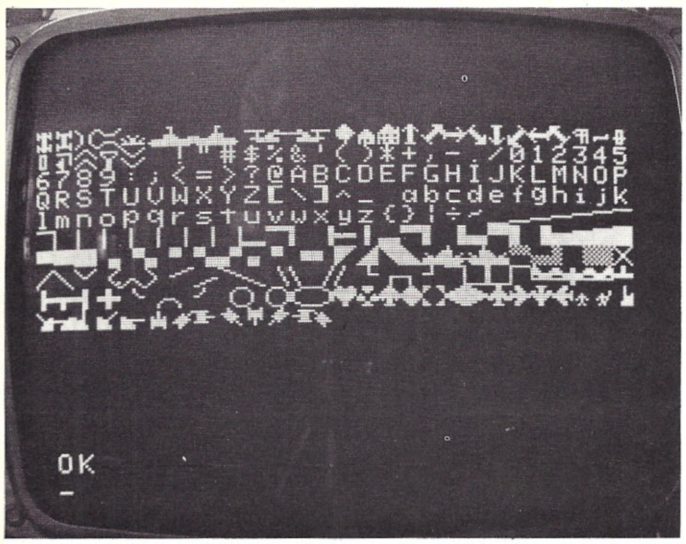


Photo 7. Sample run.

Model 620 bus adapter. The Model 620 is a connector board to connect the 610 to Ohio Scientific's 48-line bus backplane. Thus you can expand to use all of Ohio Scientific's 48-pin boards, including A/D, D/A, RS-232 port, parallel ports, PROM boards and prototype boards. All this makes it easy to expand to as big a system as you could want.

Firmware

Firmware is a read only memory (ROM), which has been programmed with software. The Superboard II has 10K of firmware in five 2K ROMs. One ROM has the system monitor, which contains input-output subroutines for a cassette, video display and keyboard.

Also contained in the ROM is a complete machine-code monitor that allows you to examine memory locations, load machine code, execute machine code programs and load machine code programs from cassette tape, but the directions do not say how to put machine code programs on the tape in the first place.

The other four ROMs contain 8K BASIC for the 6502 by Microsoft. This is a 6 1/2-digit precision BASIC with full scientific notation, trigonometry functions, string manipulations, logicals and many other useful features. Ohio Scientific claims its 8K BASIC is faster than other personal-computer BASICs. To check this out, I decided to use the benchmark programs by Rugg and Feldman published in the June 1977 *Kilobaud*. After running the programs five times each and averaging the times (I used an ordinary stopwatch), I got a total of 126.6 seconds for all seven of the benchmarks. This placed it third in a field of 25, so draw your own conclusions.

Ohio Scientific does not supply a handbook on how to program in BASIC, but only how to use their version of it. They do recommend several books on BASIC programming. What impresses me about their BASIC is that you don't have to use any spaces in the BASIC statements. This will save a lot of space-bar punching.

The Superboard II has 256 different graphics characters, including uppercase and lowercase, numbers, punctuation marks and all kinds of special graphics characters including several for use in playing games. Ohio Scientific's "The Challenger Character Graphics Reference Manual" provides an outline and decimal code for each of the 256 characters as well as a code number for each location on a 25 by 25 grid.

To put a character on the screen you merely type POKE (location code), (character code) punch RETURN, and the character appears on the screen at the location specified. "The Challenger Character Graphics Reference Manual" gives examples of how to move a character around the screen. By using an offset code, you can move the character in 12 directions on the screen.

So far Ohio Scientific has good marks for:

- fast delivery (maybe even a record)
- good packing
- good price (I paid only $279 for the board; they paid the shipping)
- good workmanship
- full-size keyboard
- 8K BASIC on ROM
- machine code monitor
- reliable and easy-to-use cassette interface
- 256 graphics characters.

Documentation

The documentation could be better. Instead of four pamphlets, there should be one user manual with better schematics. The schematics were about the worst I have seen, and I have been an electronics technician for five years. The schematics are presented as 13 separate drawings, two to a page. It is hard to trace a line that goes off the drawing. You must go through all the drawing looking for the letter and number code of that line. Since they can go to more than one page, you have to check all of the pages. Input/output lines are not labeled except by the edge-connector number. There are three exceptions to this: The microphone output of the cassette interface is labeled MIC, and the RS-232 interface is labeled RS-232 OUT and RS-232 IN. The two transistors of the RS-232 interface are not identified, except that one is an NPN and the other is a PNP.

Video Interface

The technical report and the salesman both said separate sync and video signals are available, but I could find nothing in the documentation or on the printed circuit board to back this up. This was important, because I have a "Ball Brother's" 12-inch CRT monitor that needs separate horizontal drive, vertical drive and video inputs.

In order to check the waveforms with a scope to see if they match the monitor's requirements, I had to first find the character generator, then trace the signal to the composite video output, which was labeled J2 pin 12, then trace back to where two inputs are labeled HS and VS. The vertical sync matched. The horizontal sync was inverted and too short.

By using a 74121 one-shot to invert and lengthen the horizontal sync, I received a readable display on the CRT (see Photo 1). As you can see, the display is quite readable. However, when programming and running programs, I kept getting diagonal bright spots (see Photo 2).

Using the POKE command to explore exactly how many characters I had on the screen, I discovered 27 characters per line. I also discovered an additional four characters on retrace (see Photo 3). This was allowable, since I could avoid poking characters in retrace locations, but the diagonal bright spots were not.

Examining the specifications for the monitor shows four waveforms: horizontal drive, horizontal blanking, vertical drive and vertical blanking. Since the monitor only has inputs for horizontal drive, vertical drive and video, I assumed the two blanking waveforms controlled the video reaching the monitor. I designed and built another interface (see Fig. 1) to try out this assumption. The interface worked, as you can see by comparing Photo 2 to Photo 4 and Photo 3 to Photo 5.

Photo 6 shows the listing of the program I wrote to show all of the graphics characters. Photo 7 is a run of the program. I changed the program to separate each character with a space first horizontally, then vertically and then both horizontally and vertically (Photos 8, 9 and 10).

With a normal television or monitor, you will get 24 lines by 24 characters, but since the screen is rectangular, the graphics characters are rectangular. This also happened with both of my interfaces. I did not find this acceptable. By adjusting the vertical frequency of the monitor, which stretched the display vertically, I was able to get a square graphics character. If you compare Photo 11, which was taken before I adjusted the monitor, to Photo 1 you should be able to see the difference.

Ohio Scientific could improve their video interface by having more characters per line, so that the graphics characters would look right. With 27 characters by 24 lines you have 648 total characters on the screen, but you have enough RAM for 1024 characters, so you are wasting 376 characters. Even with 30 characters by 30 lines for a total of 900 characters, you would still waste 124 characters.

Photo 1 shows the display after start-up procedure. You push four keys, BREAK, C, RETURN and RETURN, after the power is on to get into BASIC. Note the fourth line, 3327 BYTES FREE. The start-up firmware tests the RAM memory. The number 3327 tells you all of your 4K of RAM is good and you have 3327 words of memory to write programs in. The firmware uses 769 words of memory. With 8K of RAM, the screen would display 7423. If you have a program in memory and you push BREAK, to get back in BASIC without losing your program, you push W, and you are

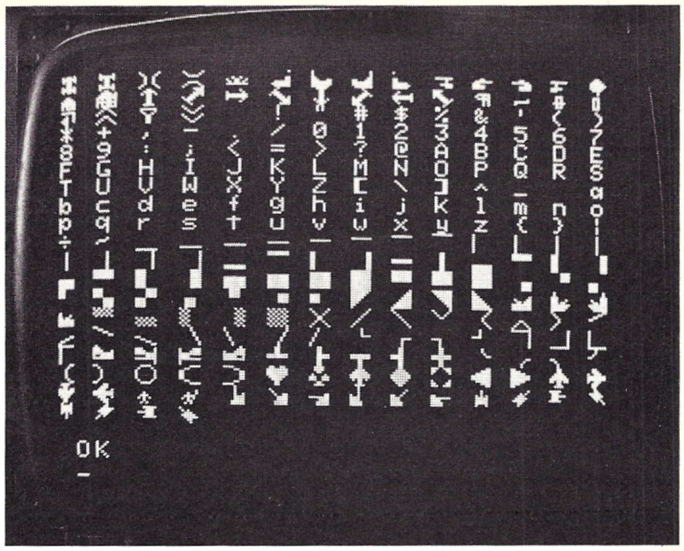


Photo 8. Program display characters separated by a horizontal space.

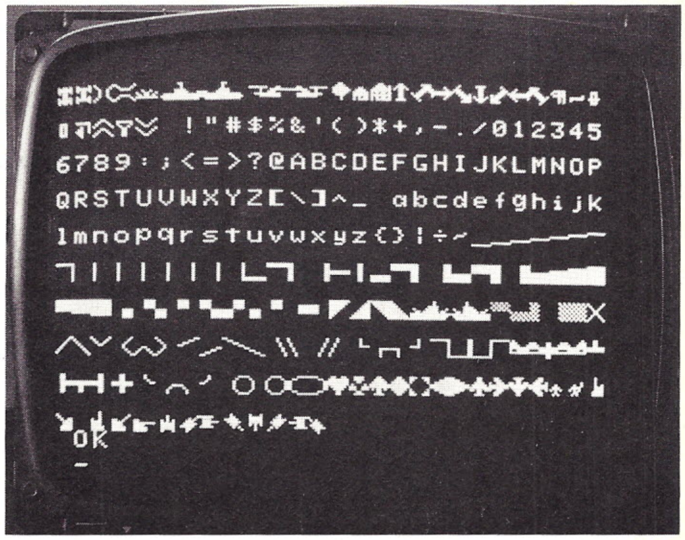


Photo 9. Program display characters separated by a vertical space.

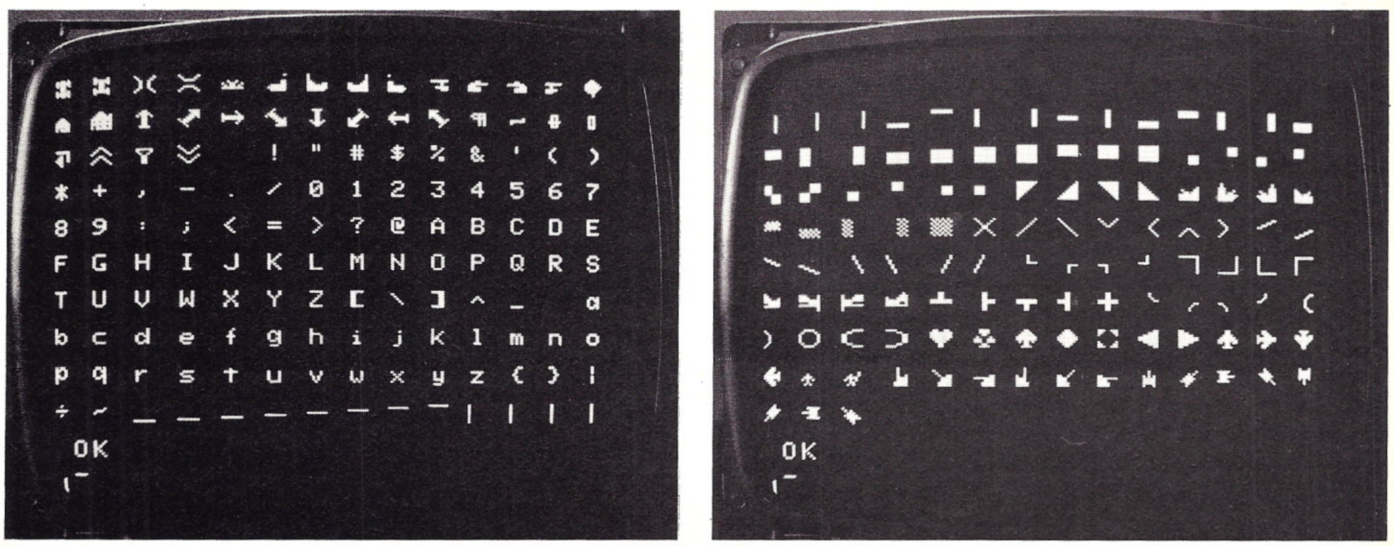


Photo 10. Program display characters separated by both horizontal and vertical spaces.

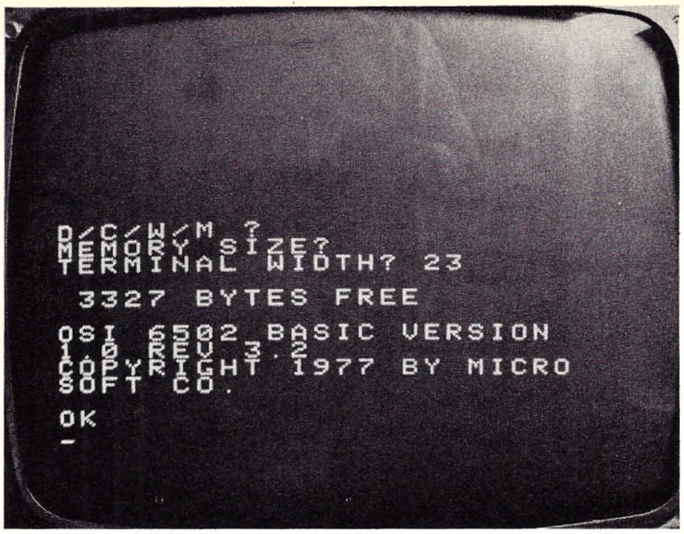


Photo 11. With a rectangular TV screen or monitor, you get rectangular graphics characters.

back in BASIC. Pushing BREAK and M puts you in the monitor so you can program in machine code.

Conclusion

Ohio Scientific's Superboard II computer system is less expensive than comparable systems, and its BASIC is fast. I think they will sell a lot of these, especially if they get them in the computer stores. I think the Ohio Scientific Superboard II is the best buy available for both beginner and expert. It is very inexpensive, especially if you already own a power supply, monitor and cassette recorder, as I did. It is also easier and less expensive to expand than other computer systems. I only wish they would supply better schematics and improve their video interface.■

References

Tom Rugg and Phil Feldman, "BASIC Timing Comparisons," *Kilobaud*, June 1977, p. 66.

"The Challenger IP Technical Report," Ohio Scientific pamphlet.

"Superboard II Challenger IP Users Manual," Ohio Scientific pamphlet.

"The 8K BASIC-in-ROM Reference Manual," Ohio Scientific pamphlet.

"The Challenger Character Graphics Reference Manual," Ohio Scientific pamphlet.

✓ Reader Service—see page 179

Len Lindsay
1929 Northport Dr.
Room 6
Madison WI 53704

Teach an Old PET New Tricks

Or vice versa.

As you probably already know, the new PET computers are incompatible with the old PETs. Memory locations are used for different things in the new PETs. Since most decent PET programs took full advantage of the PET and used its memory locations, they now will not function correctly, if at all, on the new PET.

To convert old PET programs to function on the new PET, you need a chart of the old memory location followed by the new memory location. Another chart would be needed to convert from the new locations into the old, so that old-PET users would be able to use programs coming out for the new PETs. Realizing that Commodore will probably not care to help PET users by publishing an official conversion chart, I decided to compile an unofficial conversion chart-memory map combination. Please note: Extreme care was taken to compile these charts, but *no* guarantee can be made as to their accuracy.

Rather than just list address changes, I decided to also briefly summarize what each address is used for by the PET. This is refered to as a memory map.

My sources for information included the following:

- Commodore PET user manual
- Barry Miles and Commodore Systems (in London, England), memory map
- *PET Gazette,* Spring 1979 issue, memory map by Jim Butterfield
- *Best of the PET Gazette,* memory map compiled by Roy Busdieker
- *Best of the PET Gazette,* memory map compiled by Commodore Systems in England
- "Sphinx PET Newsletter," Vol. 0, No. 4, memory map

If there is an official Commodore location description, it is listed first. Descriptions from other sources are printed inside square brackets. To fit the descriptions into one line, many words were abbreviated by dropping vowels and other letters. If two lines were needed to fully list the descriptions, the second of the two begins with a row of dots . . .

Uses of the Charts

If you have an old PET program that doesn't function correctly on your new PET, you may be able to modify it using the Old:New chart. Load the program into your new PET. Carefully scan the listing of the program. Check every PEEK and POKE used. Change the location being PEEKed or POKEd to the location listed on the chart. (Find the old location in the chart's first column. The new location is listed in the second column to its right.) In most cases this will allow the old program to function on the new PET. Save your corrected version on tape before you try a RUN.

If you have a new PET program that doesn't function correctly on your old PET, use the New:Old chart. Follow the procedure listed above.

If you are writing a PET program, you can make a version for each PET by using the charts, or you can make a program that works on both PETs. This takes a bit more effort and more memory. Your program can find out which type of PET it is running in. Simply start your program with this line:

```
1 PT = PEEK(50003)
```

An old PET will set PT = 0. A new PET will set PT = 1.

Now, for example, if you wished to clear the keyboard buffer you could code this:

```
250  IF PT THEN POKE 158,0 : GOTO 270
260  POKE 525,0
270  program continues here
```

or you could write one line to work in either PET. This is a bit more complex:

```
250  POKE 525-PT*367,0
```

In the last example, an old PET would POKE 525-0*367 or POKE 525. A new PET would POKE 525-1*367 or POKE 158.

If you want to understand what someone else's program is doing with its PEEKs and POKEs, look up the location on the applicable chart. The description on the right may help your understanding of that POKE or PEEk function.

Final Notes

These charts were compiled within two weeks of the arrival of the new PETs. If you find any flaws or areas to be clarified, please let me know. ■

Old:New chart.

```
0 OLDPET:NEW PET - POKE LOCATION CHANGE
1 -2      :1-2       USER FUNCTION ADDRESS LO, HI
3 .       :14        ACTIVE I/O CHANNEL (FOR PROMPT-SUPRESS)[CURRENT INPUT DEV#]
4 .       :"?        NULLS TO PRINT FOR CARRIAGE RETURN LINE FEED (UNUSED)
5 .       :"?        CURSOR COLUMN FOR NEXT INPUT OR PRINT
6 .       :15        TERMINAL WIDTH (UNUSED)
7 .       :16        LIMIT FOR SCANNING SOURCE COLUMNS (UNUSED)
8 -9      :17-18     LINE NUMBER BEFORE STORAGE (INTEGER ADDRESS FROM BASIC)
10 -89    :512-591   BASIC INPUT BUFFER (80 BYTES) [NUMBER OF ARRAY SUBSCRIPTS]
90 .      :3         GENERAL COUNTER FOR BASIC [SEARCH CHAR- USUALLY ':' OR ENDLIN
91 .      :4         00 USED AS DELIMITER [SCAN BETWEEN QUOTES FLAG]
92 .      :5         GENERAL COUNTER FOR BASIC [INPUT BUFFER PNTR][#OF SUBSCRIPTS]
93 .      :6         FLAG TO REMEMBER DIMENSIONED VARIABLES[1ST CHAR OF ARRAY NAME
94 .      :7         FLAG FOR VARIABLE TYPE 0= NUMERIC; 1= STRING [$FF=STRING]
95 .      :8         FLAG FOR INTEGER TAPE [80=INTEGER; 00=FLOATING POINT]
96 .      :9         FLAG TO CRUNCH RESERVED WORDS [DATA SCAN FLAG\LIST QUOTE FLAG
97 .      :10        FLAG WHICH ALLOWS SUBSCRIPTS IN SYNTAX [FN X FLAG]
98 .      :11        FLAGS INPUT OR READ [0=INPUT; 64=GET; 152=READ]
99 .      :12        FLAG SIGN OF TAN [FLAG FOR TRIG SIGNS\COMPARSN EVALUATN FLAG
100 .     :13        FLAG TO SUPPRESS OUTPUT [+=NORMAL; -=SUPRESSED]
101 .     :19        INDEX TO NEXT AVAILABLE DESCRIPTOR [VARBL DESCRPTR STACK PNTR
102 -103:20-21       POINTER TO LAST STRING TEMPORARY LO;HI [SECOND DESCRPT PONTER
104 -111:22-29       TABLE OF DOUBLE BYTE DESCRIPTORS WHICH POINT TO VARIABLES
105 .............[DESCRIPTOR STACK FOR  TEMPORARY STRINGS]
112 -113:30-31       INDIRECT ADDRESS #1 LO;HI [POINTER FOR NUMBER TRANSFER]
114 -115:32-33       INDIRRECT INDEX #2 LO;HI [NUMBER POINTER]
116 -121:34-39       PSEUDO REGISTER FOR FUNCTION OPERANDS
117 -120:"?          [PRODUCT STAGING AREA FOR MULTIPLICATION]
122 -123:40-41       POINTER TO START OF BASIC TEXT AREA LO;HI
124 -125:42-43       POINTER TO START OF VARIABLES LO;HI [END BASIC\START VARBLS
126 -127:44-45       POINTER TO ARRAY TABLE LO;HI [END VARIABLES\START ARRAYS]
128 -129:46-47       POINTER TO END OF VARIABLES LO;HI [START OF AVAILBL SPACE PTR
130 -131:48-49       START OF STRINGS POINTER LO;HI [BOTTOM OF STRINGS(MOVING DOWN
132 -133:50-51       TOP OF STRING SPACE POINTER LO;HI [MOVING DOWN]
134 -135:52-53       HIGHEST RAM ADDRESS LO;HI [TOP OF BASIC MEMORY]
136 -137:54-55       CURRENT LINE BEING EXECUTED (136=0 MEANS DIRECT)[CURRENT LIN#
138 -139:56-57       LINE NUMBER FOR CONTINUE COMMAND LO;HI [LINE# SAVED BY END]
140 -141:58-59       NEXT STATEMENT TO EXECUTE POINTER LO;HI [PREV LINE# FOR CONT]
142 -143:60-61       DATA LINE# FOR ERRORS LO;HI[LN# OF DATA LINE\CUR LN# READ PTR
144 -145:62-63       DATA STATEMENT POINTER LO;HI [READ POINTER]
145 .............[MEMORY ADDRESS OF DATA LINE
146 -147:64-65       SOURCE OF INPUT LO;HI [INPUT VECTOR(DATA ETC)][DATA STMT PNTR
148 -149:66-67       CURRENT VARIABLE NAME [CURRENT VARIABLE SYMBOLS]
```

```
150 -151:68-69    POINTER TO VARIABLE IN MEMORY LO;HI [CURRENT VARBLE START ADR
152 -153:70-71    POINTER TO VARIABLE REFERRED TO IN CURRENT FOR-NEXT
153 ..............[152 EOR OPERAND FOR WAIT\153 AND OPERAND FOR WAIT]
154 -155:72-73    POINTER TO CURRENT OPERATOR IN TABLE LO;HI
155 ..............[154 Y SAVE REGISTER\ NEW OPERATOR SAVE]
156 .   :74       SPECIAL MASK FR CURRENT OPERATOR[COMPRSN SMBL ACMLTR <1 =2 >4
157 -158:75-76    FUNCTION DEFINITION POINTER LO;HI [NUMBER WORK AREA FOR SQR
159 -160:77-78    POINTER TO A STRING DESCRIPTION LO;HI [NMBR WRK AREA 157-161]
161 .   :79       LENGTH OF ABOVE STRING
162 .   :80       CONSTANT USED BY GARBAGE COLLECT ROUTINE[3 OR 7 FOR GRBG CLCT
163 .   :81       $4C CONSTANT-6502 JMP INSTRUCTION [JUMP VECTOR FOR FUNCTIONS]
164 -165:82-83    VECTOR FOR FUNCTION DISPATCH LO;HI
166 -171:84-89    FLOATING ACCUMULATOR #3 [NUMERIC STORE AREA]
172 -173:90-91    BLOCK TRANSFER POINTER#1 LO;HI [NUMERIC STORE AREA]
174 -175:92-93    BLOCK TRANSFER POINTER#2 LO;HI [NUMERIC STORE AREA]
176 -181:94-99    FLOATING ACCUMULATOR#1 (USER FUNCTION EVALUATED HERE)
177 ..............[PRIMARY ACCUMULATR E,M,M,M,M,S][MSB PARAMETERS 181=FLPT SIGN
182 .   :100      DUPLICATE COPY OF MANTISSA OF FAC#1[TAYLOR SERIES CNST COUNTR
183 .   :101      COUNTER OF NUMBER OF BITS TO SHIFT TO NORMALIZE FAC#1[SEE 185
184 -189:102-107  FLOATING ACCUMULATOR#2 [SECONDRY ACUMULTR][DYADIC HLDNG AREA
185 ..............[183 ACCUMULATOR HIGH ORDER PROPOGATION WORD]
190 .   :108      OVERFLOW BYTE FOR FLOATING ARGUMENT [SIGN COMPARSN PRIM/SECND
191 .   :109      DUPLICATE COPY MANTISSA SIGN[LOW ORDER ROUNDNG BYTE-PRMY ACML
192 -193:110-111  POINTER TO ASCII REP OF FAC IN CONVERSION ROUTINE LO;HI
193 ..............[CASSETTE BUFFER LENGTH\ TAYLOR CONSTANT POINTER]
194 -199:112-117  CHARGET RAM CODE GETS NEXT CHARACTER FROM BASIC TEXT
200 .   :118      CHRGET RAM CODE REGETS CURRENT CHARACTERS
201 -202:119-120  POINTER TO SOURCE TEXT LO;HI
203 -223:121-140  NEXT RANDOM NUMBER IN STORAGE
224 -225:196-197  POINTER TO CURSOR LINE [SCREEN POSITION ON LINE]
226 .   :198      COLUMN POSITION OF CURSOR [POSITION OF CURSOR ON LINE][0-79]
227 -228:199-200  GENERAL PURPOSE START ADDRESS INDIRECT LO;HI [UTILITY POINTER]
228 ..............[TAPE BUFFER COUNTER, SCROLLLING][INVERSE VIDEO CURSOR=1]
229 -233:201-204  GENERAL PURPOSE AND ADDRESS DIRECT LO;HI (ALSO 180 IN NEW)
230 ..............[END OF CURRENT PROGRAM\ TAPE END ADDRESS]
231 -232..........[TAPE TIMING CONSTANTS]
233 ..............[TAPE BUFFER CHARACTER]
234 .   :205      FLAG FOR QUOTE MODE ON/OFF [DIRECT/PROGRAMMED CURSOR\0=DIRECT
235 .   :"?       [TIMER 1 INTERUPT STATUS \ 0=DISABLED]
236 .   :"?       [EOT CHARACTER RECEIVED]
237 .   :"?       [CHARACTER ERROR RECEIVED]
238 .   :209      CURRENT FILE NAME LENGTH [NUMBER OF CHARACTERS IN FILE NAME]
239 .   :210      CURRENT LOGICAL FILE NUMBER [GPIB FILE #]
240 .   :211      CURRENT PRIMARY ADDRESS [FILE COMMND(FROM OPEN)][GPIB COMMAND
241 -242:212-213  CURRENT SECONDARY ADDRES [241 DEVCE#][242 MAX LINE LNTH40/80
243 -244:214-215  START OF CURRENT TAPE BUFFER POINTER LO;HI
245 .   :216      CURRENT SCREEN LINE# [LINE WHERE CURSOR LIVES]
246 .   :217      DATA TEMPORARY FOR I/O [LAST KEY HIT(ASCII)\BUFFER CHECKSUM]
247 -248:251-252  POINTER TO START LOC FOR O.S. LO;HI [TAPE START ADDR\TAPE PNT
248 ..............[POINTER TO PROGRAM DURING VERIFY, LOAD, SAVE]
249 -250:218-219  POINTER TO CURRENT FILE NAME LO;HI [FILE NAME POINTER]
251 -254:"?       UNUSED [251 REMAINING COUNT FOR INSERT MODE]
252 ..............[252 SERIAL WORD]
253 ..............[253 NUMBER OF BLOCKS REMAINING TO WRITE]
254 ..............[254 SERIAL WORD BUFFER]
255 .   :"?       OVERFLOW BYTE THAT BASIC USES WHEN DOING FAC TO ASCII CONVRSN
256 -511:"?       62 BYTE ON BOTTOM ARE USED FOR ERROR CORRECTION IN TAPE READS
257 ..............ALSO BUFFER FOR ASCII WHEN BASIC IS EXPANDING THE FAC INTO A
258 ..............PRINTABLE NUMBER. THE REST OF PAGE 1 IS USED FOR STORAGE OF
259 ..............BASIC GOSUB AND FOR NEXT CONTEXT AND HARDWARE STACK FOR THE
260 ..............MACHINE.
261 ..............[256-266 IS BINARY TO ASCII CONVERSION AREA]
267 ..............[267-511 IS STACK AREA]
512 -514:141-143  24 HOUR CLOCK IN 1/60 SEC. [CLOCK THAT INCREMENTS 60 PER SEC
515 .   :151      KEYSTROKE VALUE [WHICH KEY DEPRSSD\255=NO KEY][MATRIX ROW-COL
516 .   :152      SHIFT FLAG (0=NO SHIFT\ 1=SHIFT) [SHIFT KEY 1 IF DEPRESSED]
517 -518:153-154  CORRECTION FACTOR FOR CLOCK LSB; MSB [CLOCK INCREMENTS 30/SEC
519 -520:249-250  INTERRUPT DRIVER FLAG FOR CASSETTE#1 SWITCHES; #2 SWITCHES
520 ..............[519 FOR CASSETTE #1 ON][520 FOR CASSETTE #2 ON]
521 .   :155      DUPLICATE OF 59410, BOTTOM ROW KEYS [KEYSWITCH PIA FLAGS]
522 .   :"?       [TIMING CONSTANT BUFFER]
523 .   :157      FLAG <> MEANS VERIFY NOT LOAD INTO MEMORY [LOAD=0\ VERIFY=1]
524 .   :150      I/O STATUS BYTE [STATUS (ST)]
525 .   :158      INDEX INTO KEYSTROKE BUFFER [# OF CHARACTERS IN KEYSTRK BUFFR
526 .   :159      FLAG TO INDICATE REVERSE FIELD ON
527 -536:623-632  INTERRUPT DRIVEN KEYSTROKE BUFFER
537 -538:144-145  IRQ RAM VECTOR LO;HI [HARDWARE INTERUPT VECTOR]
539 -540:146-147  BRK INSTRUCTION RAM VECTOR LO;HI [BREAK INTERRUPT VECTOR]
541 .   :"?       [IEEE MODE]
542 .   :"?       [END OF LINE FOR INPUT POINTR][# OF CHARACTERS ON SCREEN LINE
543 .   :"?       ?
544 -545:"?       [CURSOR LOG (ROW,COLUMN)][USED IN INPUT ROUTINE]
546 .   :"?       [PBD IMAGE FOR TAPE I/O]
547 .   :"?       [KEY IMAGE]
548 .   :"?       [0=FLASHING CURSOR: ELSE NO CURSOR SHOWS][CURSR ENABLE\POKE=0
549 .   :168      COUNTDOWN TO FLIP CURSOR [CURSOR TIMING COUNTDOWN][POKE=1]
550 .   :169      SCREEN VALUE OF CHARACTER UNDER CURSOR
551 .   :170      FLAG FOR CURSOR ON/OFF[CURSR BLINK FLAG\1=BLNK STARTED\POKE=0
552 .   :"?       [EOT BIT RECEIVED][TAPE WRITE]
553 -577:224-248  TABLE OF LSB OF START ADDRESSES OF VIDEO DISPLAY LINES(25)
554 ..............[SCREEN LINE WRAP TABLE]
578 -587:593-602  TABLE OF LOGIC ADDRESSES [LOGICAL NUMBERS OF OPEN FILES]
588 -597:603-612  TABLE OF PRIMARY ADDRESSES [DEVICE #S OF OPEN FILES]
598 -609:613-622  TABLE OF SECONDARY ADDRESSES [COMMAND/SECONDARY ADRS OPEN FILS
608 ..............[608 INPUT FROM: 0=KEYBOARD; 1=SCREEN]
610 .   :174      INDEX INTO LA, FA, SA TABLES [NUMBER OF OPEN FILES]
611 .   :175      DEFAULT INPUT DEVICE # [NORMALLY=0 FOR KEYBOARD]
612 .   :176      DEFAULT OUTPUT DEVICE#[OUTPUT TO CMD DEVCE, NORMALLY=3 SCREEN
613 .   :177      COMPUTATION OF PARITY ON CASSETTE WRITE[TAPE CHARACTER PARITY
614 .   :"?       [BYTE RECEIVED FLAG
615 .   :"?       ?
616 .   :181      TAPE BUFFER ITEM COUNTER [POINTER IN FILE NAME TRANSFER]
617 .   :"?       ?
618 .   :"?       UNUSED
619 .   :"?       ?
620 .   :"?       SERIAL BIT COUNT
621 .   :184      [COUNT OF REDUNDANT TAPE BLOCKS]
622 .   :"?       ?
623 .   :"?       [CYCLE COUNTER][FLIP FOR EVERY BIT COMING OFF OF TAPE]
624 .   :186      COUNTDOWN SYNCHRONIZATION ON CASSETTE WRITE
625 -626:187-188  INDEX NEXT CHARACTER IN/OUT TAPE BUFFER#1;#2 [625 FOR#1*626-#2
627 .   :189      COUNTDOWN SYNCHRONIZATION ON CASSETTE READ[LEADR CNTR\PASS1/2
628 .   :190      FLAG TO INDICATE BIT/BYTE TAPE ERROR [WRITE NEW BYTE]
629 .   :191      FLAG TO INDICATE TAPE ROUTINE READING SHORTS [WRITE START BIT
630 -631:192-193  INDEX TO ADDRESSES TO CORRECT ON TAPE READ PASS 1; PASS 2
631 ..............[631 TAPE DROPOUT CNTR][630 PASS1 ERR PNTR\631 PASS2 ERR PNTR
632 .   :194      FLAG FOR CASSETE READ-TELLS CURRENT FUNCTN-COUNTDWN,READ[*635
633 .   :195      COUNT OF SECONDS OF SHORTS TO WRITE BEFORE DATA [CHECKSUM]
634 -825:634-825  BUFFER FOR CASSETTE #1 (192 BYTES)
635 (CONTINUED 632) [632 READ MODE: 0=SCAN 1-15=COUNTDOWN $40=LOAD $80=END]
636 ..............[632 TAPE CORRECTION COUNT]
826 -1017:826-1017  BUFFER FOR CASSETTE #2 (192 BYTES)
READY.
```

New:Old chart.

```
0 NEWPET/OLDPET- POKE LOCATION CHANGE
1 -2    :1-2      USER FUNCTION ADDRESS LO;HI
3 .     :90       GENERAL COUNTER FOR BASIC [SEARCH CHRTR-USUALLY ':' OR ENDLIN
4 .     :91       ;00 USED AS DELIMITER [SCAN BETWEEN QUOTES FLAG]
5 .     :92       GENERAL COUNTER FOR BASIC [INPUT BUFFR PNTR][#OF SUBSCRIPTS]
6 .     :93       FLAG TO REMEMBER DIMENSIONED VARIABLES [1ST CHAR OF ARRAY NAM
7 .     :94       FLAG FOR VARIABLE TYPE 0= NUMERIC; 1= STRING [$FF=STRING]
8 .     :95       FLAG FOR INTEGER TAPE [80=INTEGER; 00=FLOATING POINT]
9 .     :96       FLAG TO CRUNCH RESERVED WORDS [DATA SCAN FLAG\LIST QUOTE FLAG
10 .    :97       FLAG WHICH ALLOWS SUBSCRIPTS IN SYNTAX [FN X FLAG]
11 .    :98       FLAGS INPUT OR READ [0=INPUT; 64=GET; 152=READ]
12 .    :99       FLAG SIGN OF TAN [FLAG FOR TRIG SIGNS\COMPARSN EVALUATN FLAG
13 .    :100      FLAG TO SUPPRESS OUTPUT [+=NORMAL; -=SUPRESSED]
14 .    :3        ACTIVE I/O CHANNEL (FOR PROMPT-SUPRESS)
15 .    :6        TERMINAL WIDTH (UNUSED)
16 .    :7        LIMIT FOR SCANNING SOURCE COLUMNS (UNUSED)
17 -18  :8-9      LINE NUMBER BEFORE STORAGE (INTEGER ADDRESS FROM BASIC)
19 .    :101      INDEX TO NEXT AVAILABLE DESCRIPTOR[VARIBL DESCRPTR STACK PNTR
20 -21  :102-103  POINTER TO LAST STRING TEMPORARY LO;HI [SECOND DESCRPT PONTER
22 -29  :104-111  TABLE OF DOUBLE BYTE DESCRIPTORS WHICH POINT TO VARIABLES
23 ...............[DESCRIPTOR STACK FOR  TEMPORARY STRINGS]
30 -31  :112-113  INDIRECT ADDRESS #1 LO;HI [POINTER FOR NUMBER TRANSFER]
32 -33  :114-115  INDIRRECT INDEX #2 LO;HI [NUMBER POINTER]
34 -39  :116-121  PSEUDO REGISTER FOR FUNCTION OPERANDS
40 -41  :122-123  POINTER TO START OF BASIC TEXT AREA LO;HI
42 -43  :124-125  POINTER TO START OF VARIABLES LO;HI [END OF BASIC\START VARBL
44 -45  :126-127  POINTER TO ARRAY TABLE LO;HI [END VARIABLES\START ARRAYS]
46 -47  :128-129  POINTER TO END OF VARIABLES LO;HI [START OF AVAILBL SPACE PTR
48 -49  :130-131  START OF STRINGS POINTER LO;HI [BOTTOM OF STRINGS(MOVING DOWN
50 -51  :132-133  TOP OF STRING SPACE POINTER LO;HI [MOVING DOWN]
52 -53  :134-135  HIGHEST RAM ADDRESS LO;HI [TOP OF BASIC MEMORY]
54 -55  :136-137  CURRENT LINE BEING EXECUTED (54=2 MEANS DIRECT) [CURRENT LN#
56 -57  :138-139  LINE NUMBER FOR CONTINUE COMMAND LO;HI [LINE# SAVED BY END]
58 -59  :140-141  NEXT STATEMENT TO EXECUTE POINTER LO;HI [PREV LINE# FOR CONT]
60 -61  :142-143  DATA LINE# FOR ERRORS LO;HI [LINE# OF DATA LINE]
62 -63  :144-145  DATA STATEMENT POINTER LO;HI [READ POINTER]
63 ...............[MEMORY ADDRESS OF DATA LINE
64 -65  :146-147  SOURCE OF INPUT LO;HI [INPUT VECTOR (DATA ETC)][DATA STMT PTR
66 -67  :148-149  CURRENT VARIABLE NAME [CURRENT VARIABLE SYMBOLS]
68 -69  :150-151  POINTER TO VARIABLE IN MEMORY LO;HI [CURRENT VARBLE START ADR
70 -71  :152-153  POINTER TO VARIABLE REFERRED TO IN CURRENT FOR-NEXT
72 -73  :154-155  POINTER TO CURRENT OPERATOR IN TABLE LO;HI
73 ...............[154 Y SAVE REGISTER\ NEW OPERATOR SAVE]
74 .    :156      SPECIAL MASK FOR CURRENT OPERATOR [CMPR SYMBL ACMLTR <1 =2 >4
75 -76  :157-158  FUNCTION DEFINITION POINTER LO;HI [NUMBER WORK AREA FOR SQR..
77 -78  :159-160  POINTER TO A STRING DESCRIPTION LO;HI [NMBR WRK AREA 157-161]
79 .    :161      LENGTH OF ABOVE STRING
80 .    :162      CONSTANT USED BY GARBAGE COLLECT ROUTINE [3 OR 7 FOR GRBG CLT
81 .    :163      $4C CONSTANT-6502 JMP INSTRUCTION [JUMP VECTOR FOR FUNCTIONS]
82 -83  :164-165  VECTOR FOR FUNCTION DISPATCH LO;HI
84 -89  :166-171  FLOATING ACCUMULATOR #3 [NUMERIC STORE AREA]
90 -91  :172-173  BLOCK TRANSFER POINTER#1 LO;HI [NUMERIC STORE AREA]
92 -93  :174-175  BLOCK TRANSFER POINTER#2 LO;HI [NUMERIC STORE AREA]
94 -99  :176-181  FLOATING ACCUMULATOR#1 (USER FUNCTION EVALUATED HERE)
95 ...............[PRIMARY ACCUMULATR E,M,M,M,M,S][MSB PARAMETRS 181=FLPT SIGN
100 .   :182      DUPLICATE COPY OF MANTISSA OF FAC#1 [TAYLOR SERIES CONST CNTR
101 .   :183      COUNTER - NUMBER OF BITS TO SHIFT TO NORMALIZE FAC#1 [SEE 103
102 -107:184-189  FLOATING ACCUMULATOR#2 [SECONDRY ACUMULTR][DYADIC HLDNG AREA
103 ..............[101 IS ACCUMULATOR HIGH ORDER PROPOGATION WORD]
108 .   :190      OVERFLOW BYTE FOR FLOATING ARGUMENT [SIGN COMPARISN PRIM/SCND
109 .   :191      DUPLICATE COPY MANTISSA SIGN [LOW ORDER ROUNDNG BYTE-PRM ACML
110 -111:192-193  POINTER TO ASCII REP OF FAC IN CONVERSION ROUTINE LO;HI
111 ..............[CASSETTE BUFFER LENGTH\ TAYLOR CONSTANT POINTER]
112 -117:194-199  CHARGET RAM CODE GETS NEXT CHARACTER FROM BASIC TEXT
118 .   :200      CHRGET RAM CODE REGETS CURRENT CHARACTERS
119 -120:201-202  POINTER TO SOURCE TEXT LO;HI
121 -140:203-223  NEXT RANDOM NUMBER IN STORAGE \ OR
136 -140 .........NEXT RANDOM NUMBER IN STORAGE
141 -143:512-514  24 HOUR CLOCK IN 1/60 SEC [CLOCK THAT INCREMENTS 60 PER SEC
144 -145:537-538  IRQ RAM VECTOR LO;HI [HARDWARE INTERUPT VECTOR]
146 -147:539-540  BRK INSTRUCTION RAM VECTOR LO;HI [BREAK INTERRUPT VECTOR]
148 .   :"?       NMI RAM VECTOR
150 .   :524      I/O OPERATION STATUS BYTE [STATUS (ST)]
151 .   :515      LAST KEY INDEX [WHICH KEY DEPRSD\255=NO KEY][MATRIX ROW-COL
152 .   :516      SHIFT FLAG (0=NO SHIFT\ 1=SHIFT) [SHIFT KEY 1 IF DEPRESSED]
153 -154:517-518  CORRECTION FACTOR FOR CLOCK LSB; MSB [CLOCK INCREMENTS 30/SEC
155 .   :521 "    DUPLICATE OF 59410(NEW=?)BOTTOM ROW KEYS [KEYSWITCH PIA FLAGS
157 .   :523      FLAG <> MEANS VERIFY NOT LOAD INTO MEMORY [LOAD=0*VERIFY=1]
158 .   :525      INDEX INTO KEYSTROKE BUFFER [# OF CHARACTERS IN KYSTRK BUFFER
159 .   :526      FLAG TO INDICATE REVERSE FIELD ON
160 -166:"?       UNUSED
167 .   :"?       CURSOR ON FLAG
168 .   :549      COUNT OF JIFFIES TO BLINK CURSOR[CURSOR TIMING CNTDN][POKE=1]
169 .   :550      SCREEN VALUE OF CHARACTER UNDER CURSOR
170 .   :551      CHAR SAVED DURING BLNK\CURSR ON/OFF FLG[CURSR BLNK FLG\POKE=0
171 -173:"?       UNUSED
174 .   :610      POINTER INTO LOGICAL FILE TABLE [NUMBER OF OPEN FILES]
175 .   :611      DEFAULT INPUT DEVICE # [NORMALLY=0]
176 .   :612      DEFAULT OUTPUT DEVICE # [OUTPUT TO CMD DEVICE, NORMALLY=3]
177 .   :613      TAPE VERTICAL PARITY\COMPUTATION OF PARITY ON CASSETTE WRITE
178 -185:"?       UNUSED [?]
180 .   :229-233  GENERAL PURPOSE AND ADDRESS DIRECT LO;HI (ALSO LOCATN 201-204
```

```
181 .   :616       TAPE BUFFER ITEM COUNTER [POINTER IN FILE NAME TRANSFER]
184 .   :621       [COUNT OF REDUNDANT TAPE BLOCKS]
186 .   :624       SYNC ON TAPE HEADER COUNT\COUNTDOWN SYNC ON CASSETTE WRITE
187 -188:625-626   POINTER TO ACTIVE CASSETTE \ OR
188 ..............INDEX NEXT CHARACTER IN/OUT TAPE BUFFER#1;#2[187 FOR#1\188-#2
189 .   :627       COUNTDOWN SYNCHRONIZATION - CASSETTE READ[LEADER CNTR\PASS1/2
190 .   :628       FLAG TO INDICATE BIT/BYTE TAPE ERROR [WRITE NEW BYTE]
191 .   :629       FLAG TO INDICATE TAPE ROUTINE READING SHORTS [WRITE START BIT
192 -193:630-631   INDEX TO ADDRESSES TO CORRECT ON TAPE READ PASS 1; PASS 2
193 ..............[192 FOR PASS 1 ERROR LOG POINTR\193 FOR PASS 2 ERR LOG PNTR
194 .   :632       FLAG-CASSETTE READ-TELLS CURRENT FUNCTN-COUNTDWN,READ
195 .   :633       COUNT OF SECONDS OF SHORTS TO WRITE BEFORE DATA [CHECKSUM]
196 -197:224-225   POINTER TO CURSOR POSITION [SCREEN POSITION ON LINE]
198 .   :226       COLUMN POSITION OF CURSOR [POSITION OF CURSOR ON LINE][0-79]
199 -200:227-228   LOAD START ADDRESS LO;HI [UTILITY POINTER]
200 ..............[TAPE BUFFER,SCROLLLING][INVERSE VIDEO CURSOR=1]
201 -202:229-233   PRINT LOAD END ADDRESSS LO;HI (INCLUDES LOCATION 180)
202 ..............[END OF CURRENT PROGRAM\ TAPE END ADDRESS]
205 .   :234       FLAG FOR QUOTE MODE ON/OFF [DIRECT/PROGRAMMED CURSOR*0=DIRECT
206 -208:"?        UNUSED
209 .   :238       CURRENT FILE NAME LENGTH [NUMBER OF CHARACTERS IN FILE NAME]
210 .   :239       CURRENT FILE LOGICAL ADDRESS [GPIB FILE #]
211 .   :240       CURRENT PRIMARY ADDRESS [FILE COMMAND(FROM OPEN)][GPIB COMND]
212 -213:241-242   CURRENT SECONDARY ADDRES [212 DEVCE#][213 MAX LINE LNTH40/80
214 -215:243-244   START OF CURRENT TAPE BUFFER POINTER LO;HI
216 .   :245       CURRENT SCREEN LINE# [LINE WHERE CURSOR LIVES]
217 .   :246       DATA TEMPORARY FOR I/O [LAST KEY HIT(ASCII)*BUFFER CHECKSUM]
218 -219:249-250   POINTER TO CURRENT FILE NAME LO;HI [FILE NAME POINTER]
220 -221:"?        UNUSED
222 .   :"?        CASSETTE READ BLOCK COUNT
223 .   :"?        UNUSED
224 -248:553-577   TABLE OF LSB OF START ADDRESSES OF VIDEO DISPLAY LINES(25)
225 ..............[SCREEN LINE WRAP TABLE]
249 -250:519-520   INTERRUPT DRIVER FLAG FOR CASSETTE#1 SWITCHES; #2 SWITCHES
250 ..............[249 FOR CASSETTE #1 ON][250 FOR CASSETTE #2 ON]
251 -252:247-248   POINTER TO START LOC FOR O.S. LO;HI [TAPE START ADDR*TAPE PNT
252 ..............[POINTER TO PROGRAM DURING VERIFY, LOAD]
512 -591:10-89     BASIC INPUT BUFFER (80 BYTES) [NUMBER OF ARRAY SUBSCRIPTS]
513 ..............512-513 IS THE PROGRAM COUNTER
514 ..............514 IS PROCESSOR STATUS
515 ..............515 IS ACCUMULATOR
516 ..............516 IS X INDEX
517 ..............517 IS Y INDEX
518 ..............518 IS STACK POINTER
519 -520..........519-520 IS USER MODIFIABLE IRQ
593 -602:578-587   LOGICAL FILE NUMBERS [LOGICAL NUMBERS OF OPEN FILES]
603 -612:588-597   PRIMARY DEVICE NUMBERS [DEVICE #S OF OPEN FILES]
613 -622:598-609   TABLE OF SECONDARY ADDRESSES [COMMAND/SECONDARY ADRS OPEN FIL
623 -632:527-536   INTERRUPT DRIVEN KEYSTROKE BUFFER
634 -825  BUFFER FOR CASSETTE #1 (192 BYTES)
826 -1017:826-1017  BUFFER FOR CASSETTE #2 (192 BYTES)
READY.
```

A Circular Handle on Graphics

This is another of Allan's handles on programming, which started running last month.

Allan S. Joffe
1005 Twining Rd.
Dresher PA 19025

The circle is an interesting graphics form that can enlarge your programming skills. It is probably the oldest graphics symbol that man has invented. Originally, the circle had no name, and if someone wanted to describe a circle, he or she merely used arm or finger motions. This sad state of affairs persisted into the Greek age of Enlightenment. It was the task of two Greek tailors (who were amateur math buffs), named Euripides and Ifixedes, to finally give this graphics symbol a name. They said, "Let's call it a circle," and the name has stuck to this very day.

What Is a Circle?

A circle is the locus of all points equidistant from a single central point (cleverly called the center). Fig. 1 shows the circle with two diameters drawn at right angles to each other and also features a right triangle inscribed within the circle. The radius, R, is the hypotenuse, the altitide is Y, and a segment of the diameter forms the base of the triangle. Point B is the intersection of both R and Y with the circumference. It is easy to see that if we draw another triangle as represented by the dotted line, R will remain constant, but X and Y will vary and in so doing move the point B around the circle.

Using the Pythagorean theorem (Pythagoras had all his togas made by Euripides and Ifixides), we can determine an expression that will give us the Y value for any position of the radius. What we are attempting to do here is determine enough facts from the problem at hand to produce a program that will graph our circle.

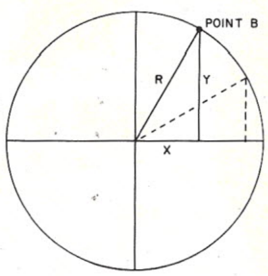


Fig. 1.

Our expression becomes $Y = \sqrt{R^2 - X^2}$.

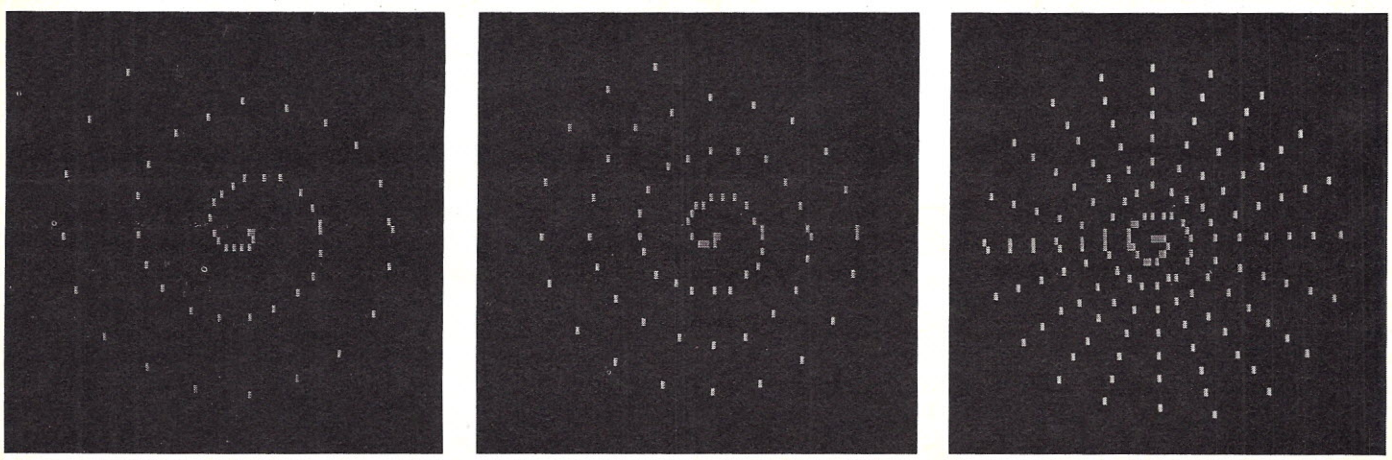
The Program

To create our program we need two variables, X and Y. X is created by stepping off equal incremental values that must range between $-R$ and R. Essentially we are creating radial points equidistant from the circle center. To create the Y points we have to set up an expression that will really set up a series of inscribed triangles, moving R around the circle and solving the various triangles for the Y values that exist at any given moment. Then we set these points using the built-in graphics capabilities of the TRS-80.

The program must also take into account the need for scaling the produced values to stay within the graphics bounds and must contain horizontal and vertical centering factors to set the center of the circle at the center of the screen. Remember, the TRS-80 X = 0 and Y = 0 point is at the upper left-hand corner of the screen, and that is no place for the center of any self-respecting circle!

The program is written in TRS-80 BASIC Level II. In Program A line 60 may be deleted with no effect on the running of the program. It was inserted for emphasis rather than need.

The value for R in line 10 was selected to produce the largest possible circle that the graphics limits will allow. Reducing this value will reduce the diameter of the graphed circle. Lines 40 and 90 contain the necessary factors to center the circle on the TRS-80 screen, and their derivation should be obvious. Line 110 is merely a holding loop so the final display will not be busted by the appearance of READY after the program has run. I suggest that for exploratory purposes you insert a line 55 END, which will allow you to see just how



Spiral variations formed by changing the pitch control (line 60 in Program B). *(Photos by WA3PTC)*

```
5    CLS
10   R = 39
20   FOR X = - R TO R STEP 4
30   Y = SQR (R * R - X * X)/2
40   SET (X + 57, Y + 23)
50   NEXT X
60   R = 39
70   FOR X = R TO - R STEP -4
80   Y = SQR (R * R - X * X)/2
90   SET (X + 57, -Y + 23)
100  NEXT X
110  GOTO 110
```

Program A. Circle generation.

```
5    CLS
10   L = 20/2/3.14159
20   FOR B = 0 TO 10
30   FOR T = 0 TO 20
40   Y = K * SIN (T/L)
50   X = K * COS (T/L)
60   K = K + .3
70   X = X + 60
80   Y = (Y + 46)/2
90   IF Y < 0 GOTO 120
100  SET (X,Y)
110  NEXT T: NEXT B
120  GOTO 120
```

Program B. Spiral generation.

each half of the program operates.

It is worth noting that in lines 30 and 80 we used R * R and X * X rather than R ↑ 2 and X ↑ 2. If you time the formation of this circle as per Program A and then reprogram using the R ↑ 2 method, you will find that this latter approach will double the time taken to graph the circle. If you are inclined to view speed as a factor, this holds true for any such programming usage.

Modifications

As you have probably guessed, it would not be much of a trick to change the listing to produce concentric circles on the screen. (See Example 1.)

These changes will set a series of concentric circles but will complete the bottoms of all circles and then the tops of all circles in order. If you wish something fancier, then we have to shift gears a bit and make a few more changes to Program A. (See Example 2.)

Each time the program goes through the loop, the incremented value of the counter C will be added to the radius, R. This will produce a series of concentric circles, each of which will be completed fully, one at a time.

Line 15 is necessary to limit the size of the outermost circle so it stays within the TRS-80 graphics limits. If you modify Line 110 to make it read 110 GOTO 5, after each circle is formed, it will be blanked and then the next larger circle will form on the screen. With this much under our collective belts we can push back the frontier of the circle's disoriented cousin, the spiral.

Spiral

A disoriented circle, namely the spiral, may be thought of as a circle that couldn't stand to keep its diameter constant. This historic commentary was made by Sir William Davol, a sixteenth century Englishman of noted obscurity. His only other contribution to mathematics was the cogent observation that "Two fifths make one cavort," which was a commentary on the math of his era.

Program B gives us the basic program to produce the spiral. This program was generated by one of my children, Dan.

Now for a few comments on certain controlling elements within the program. Line 20 will give you basic control over the number of turns or convolutions in the spiral. Line 70 centers the image horizontally on the screen. Line 80 performs the same function for the vertical dimension. Line 60 controls the pitch or displacement of one turn from the next, within the spiral. In this context it will interact somewhat with line 20, primarily due to the graphics screen's discrete limitations.

Mastering and understanding the various programming tactics used to form these two geometric figures will, I hope, spur you on to greater heights. It can definitely lead you one step down the road to becoming the master of your TRS-80.■

Line 10 becomes FOR R = 15 to 39 STEP 4
Line 50 becomes NEXT X,R
Line 60 becomes FOR R = 15 to 39 STEP 4
Line 100 becomes NEXT X,R

Example 1.

Line 10 becomes	R = 15
Add new line 14	C = C + 2
Add new line 15	IF C >28 THEN GOTO 120
Add new line 16	R = R + C
Line 50 becomes	next X
Line 60 becomes	R = 15
Add new line 64	C = C + 2
Add new line 66	R = R + C
Line 100 becomes	next X
Line 110 becomes	GOTO 10
Add new line 120	GOTO 120

Example 2.

✓ Reader Service—see page 179

R. W. Petty
21205 Roscoe #56
Canoga Park CA 91304

1802 PILOT

This version of PILOT—PILOT is a programming language that has been developed for computer-aided instruction—is designed to work on small systems that utilize the 1802.

PILOT is a programming language designed for use in computer-aided instruction. It is widely used in educational institutions, and a number of interpreters are available for various microprocessors. This version is designed to run on small 1802 systems. It is a complete software package with full capability for the creation and execution of PILOT programs and has adequate power to run most published PILOT programs.

A simple PILOT program and a sample run accompany the article. Since they use most of the available instructions, you can refer to them as you read the following description of 1802 PILOT. Also, there is a good description of a similar version of PILOT in the March 1978 issue of *Kilobaud* ("Programmed Instruction Made Easy: Tiny PILOT," p. 66, by Allen S. Krieger).

1802 PILOT Instructions

Table 1 lists the instructions and their syntax. For the most part the instructions correspond to other versions of PILOT; however, some details differ.

The main appeal of PILOT as a language is the relative ease of writing programs that allow conversations between the user and the computer. Central to this concept are the TYPE instruction, which allows the computer to "speak," and the ASK instruction, which permits the user his say. The instructions allow for the exchange of numbers (numeric variables) and words or phrases (string variables).

Of course, if the computer is to carry on this conversation with any degree of intelligence, the program must be able to "understand" whatever the user says. This understanding is provided by the MATCH and EXAMINE instructions. The MATCH instruction allows the program to check for the occurrence of key words or phrases within the user's input. Likewise, the EXAMINE instruction allows the program to test the status of numbers.

Having determined that certain events have or have not occurred, the program then makes use of the YES and NO instructions to control its responses. The JUMP instruction is available for branching to different sections of the program, and the USE/RETURN instructions allow access to subroutines.

Although PILOT is mainly concerned with text material, the ability to manipulate numbers is provided with the COMPUTE statement. Also, the RANDOM Z instruction is avail-

TYPE—T: (text) (#n) ($n) (;)—Outputs specified text, numerical values and contents of string variables to terminal. Semicolon at end of line suppresses CR/LF.
ASK—A: (#n or $n)—Requests input from terminal. Stores response if numeric or string variable is specified.
MATCH—M: string (,string)—Compares response obtained with ASK with one or more strings. If match occurs, FLAG is set to YES, else FLAG is set to NO.
EXAMINE—X: exp (< = > exp)—Sets FLAG to YES if exp evaluates to 0. If comparison is specified then FLAG is set to YES if condition is satisfied. FLAG set to NO otherwise.
YES—_Y: xxxx—Instruction (_xxxx means any valid PILOT statement) is performed if FLAG is set to YES.
NO—_N: xxxx—Instruction is performed if FLAG is set to NO.
JUMP—J: exp—Causes program branch to statement with same label number as the value of exp.
USE:—U: exp—Call to subroutine with label number equal to value of exp. Nesting allowed to 24 deep.
RETURN—R:—Returns subroutine (USE).
COMPUTE—C: n = exp—Assigns the value of exp to numeric variable n.
RANDOM Z—Z: exp—Assigns a random number with a value of 0 to (exp − 1) to numeric variable Z.
CONTROL—K: exp—Outputs the ASCII equivalent of the value of exp to the terminal (i.e., 13 gives CR, 10 gives LF).
END—E:—Terminates execution of program.
Definitions

ccc—Constant 0 to 255.
%ccc—Statement label. May precede any PILOT statement. Statements need not be labeled. Statements need not be labeled in sequence.
n—Capital letters A through Z (variable names).
#n—Numeric variable in TYPE and ASK.
$n—String variable in TYPE and ASK.
( )—Denotes optional part of statement.
*exp—An expression consisting of constants and numeric variables with operators + − * /. For example:*

*A + B − 10 * 2/C*

Expressions are evaluated from left to right, thus the above expression is equivalent to:

((A + B − 10) 2)/C*

Table 1. 1802 PILOT instructions.

UP (Uccc)—Moves text pointer up ccc lines. Displays new current line.
DOWN (Dccc)—Moves text pointer down ccc lines. Displays new current line.
BEGIN (B)—Moves text pointer up to start of text. Displays new current line.
END (E)—Moves text pointer down to end of text.
INSERT (I text)—Inserts one line of text in front of current line.
KILL (Kccc)—Deletes ccc lines of text starting with current line. Displays new current line.
WRITE (Wccc)—Writes ccc lines of text to terminal starting with current line. Position of text pointer is not changed.
CLEAR (C)—Deletes all text.
STORE (S)—Outputs entire text to cassette.
LOAD (L)—Inputs text from cassette. Appends to existing text.
MONITOR (M)—Returns control to system monitor.
PILOT (P)—Causes execution of PILOT program.

Table 2. 1802 editor commands.

```
0000 F800 B3B4 B5F8 09A3 D3F8 08B2 F8FF A2F8;
0010 23A4 F839 A5D4 004A D400 5FD4 008A C001;
0020 809F D3BF E296 7386 7397 7387 7393 B683;
0030 A646 B346 A330 219F D3BF E212 96B3 86A3;
0040 72A7 72B7 72A6 F0B6 3037 F808 B7F8 5FA7;
0050 F800 2757 873A 50D5 D400 5FF8 035A D5F8;
0060 09BA F800 AAF8 025A 1AD5 D481 A4D5 D481;
0070 3ED5 46D4 006A 06F8 033A 7216 D5D4 0076;

0080 0D0A 0000 0000 0000 03D5 D400 7DD4 0076;
0090 4544 4954 03D4 007D D53B A39A FC01 BA30;
00A0 F808 B7F8 00A7 4657 D5D4 00B4 381B 0BFB;
00B0 2032 ADD5 F808 BBF8 60AB D5A7 F808 B707;
00C0 A9D5 F800 A889 B9F8 64D4 00D4 F80A D400;
00D0 D499 30E2 A999 B7D4 00E8 893A E088 32E7;
00E0 1889 F930 D400 6AD5 8952 32F8 F800 A997;
00F0 3819 F733 F1F4 B9D5 D400 A001 D50D 3AF9;
0100 F808 B746 A79F 57D5 D400 B4F8 00A7 D400;
0110 6E5B FB08 3242 FB10 3250 FB15 3259 F804;

0120 322E 1787 FB3E 3259 4BD4 006A 300E F820;
0130 5B17 1BD4 006A 87FB 3E32 5987 FA03 320E;
0140 302E 8732 0E27 2BD4 0076 1315 0003 300E;
0150 D400 760D 1500 0330 08F8 0D5B D400 7DD5;
0160 5207 FB03 327A 47F3 326E 1717 3061 F801;
0170 B8F8 75A8 D847 B347 A3D3 D400 A002 D5FF;
0180 F800 D400 BB32 96D4 00C2 F83F D400 6AD4;
0190 007D D400 A000 F83E D400 6AD4 0108 D400;
01A0 A9F8 0137 F8AD A748 D401 6030 8042 0200;
01B0 4300 5844 02D0 4502 1A49 0238 4B03 394C;

01C0 0286 4D80 0050 03F9 5302 6755 02B6 5702;
01D0 A003 0303 0303 0303 0AFF 413B E50A FF5B;
01E0 33E5 F800 D5F8 01D5 F820 5818 F80E AFF8;
01F0 3B58 18F8 2058 182F 8F3A F3D5 D400 6C3B;
0200 D400 5F9A B88A A808 FB03 3216 FB0E 3216;
0210 48D4 006A 3007 D400 7DD5 D402 2003 03D5;
0220 46B7 46A7 EAF8 03F3 3235 87F3 3235 97F3;
0230 3237 1A30 25F8 01D5 EA9A B78A A74A FB03;
0240 3A3D 9AFB 2032 5E07 52F8 0257 2A72 73FB;
0250 023A 4C17 0257 4B5A 1AFB 0D3A 3DD5 2AF8;

0260 035A D400 A003 D5D4 005F 4AB7 F8FF A727;
0270 873A 6F97 D481 A497 FB03 3284 FB0E 3A6A;
0280 F80A 306B 2AD5 D402 2003 0338 1A9A FB20;
0290 325E D481 40FB 0A32 8D9F 5AFB 033A 8CD5;
02A0 D403 1F32 B59A B88A A8D4 0207 48FB 0332;
02B0 B529 893A A9D5 D403 1F32 CC2A 0AFB 0232;
02C0 CB2A 0AFB 0D3A BC29 893A BB1A D402 03D5;
02D0 D403 1F32 D8D4 02DC D402 03D5 4AFB 0332;
02E0 EAFB 0E3A DC29 893A DC38 2AD5 1D2C 8C3A;
02F0 EAF8 0D5D D5F8 09BD D404 859A BF8E 73D4;

0300 B9FF 303B 1C99 FF3A 331C 99FA 0FB9 89FE;
0310 FE52 89F4 FE52 99F4 A9F8 00D5 F801 D5D4;
0320 00B4 F800 A9A7 3817 4BD4 0300 3227 873A;
0330 3799 FB0D 3A28 1989 D5D4 031F 324C 9AB7;
0340 8AA7 D402 DCD4 0350 97BA 87AA D402 03D5;
0350 4A57 17FB 033A 50D5 203B 79D4 011B 2D20;
0360 381A 0AFB 0D32 79FB 2932 8DFB 0732 7DFB;
0370 1832 7C0A D400 6A30 61D4 007D D51A 0AD4;
0380 01D8 3AC2 0AD4 00BB D400 C230 611A 9AB7;
0390 8AA7 D403 AD3A C81A 0AFB 0D32 A7FB 2732;

03A0 A70A D400 6A30 9797 BA87 AA30 61D4 0220;
03B0 0303 1AD4 0220 2403 3AC1 0752 1A0A F33A;
03C0 B2D5 D400 A004 3079 D400 A005 3079 FC07;
03D0 33D6 FC0A 33CC 4104 9043 0690 4505 104A;
03E0 0520 4D05 4D52 0539 5403 6055 0524 5806;
03F0 0A5A 05A0 4B05 C803 D5D4 005F D400 4AD4;
0400 0478 D402 2003 031A F803 5AD4 0481 1AD4;
0410 0220 3A03 3A53 D404 782A 0AFB 2032 190A;

0420 FB4E 3260 0AFB 5932 570A B9D4 0481 1AF8;
0430 03B7 F8D6 A799 D401 60F8 00D4 00BB 320B;
0440 D404 8138 2A0A FB02 324F 0AFB 0D3A 441A;
0450 D402 03D4 008A D5F8 01D4 00BB 320B 3019;
0460 F801 D400 BB32 1930 0BF8 02D4 003B FE52;
0470 F810 F4A8 F808 B8D5 D404 699A 5818 8A58;
0480 D5D4 0469 48BA 08AA D509 AA1A 757D 6FBC;
0490 F83F D400 6AD4 0108 D402 2024 0D32 B9D4;
04A0 0481 D402 2023 0D32 AAD5 1AD4 031F 0AD4;
04B0 01D8 3AF9 0AD4 0503 D51A 9AB7 8AA7 07D4;

04C0 01D8 3AFE 07B8 D403 AD3A E39A B79A A727;
04D0 0AFB 0332 DCFB 2732 DC1A 30D0 D403 5097;
04E0 BA87 AAD4 0220 0303 F824 5A1A 985A 1AF8;
04F0 035A D400 A9D4 0238 D5D4 00A0 06D5 D400;
0500 A007 D5A7 F808 B789 57D5 480E A7EC FBBF;
0510 D402 2003 03D4 0478 D58A 23A7 EEA2 7D1E;
0520 D407 00D5 F802 D400 BBA7 1787 D401 0002;
0530 D407 0032 38D4 0539 D5F8 02D4 00BB 3248;
0540 A727 87D4 0100 02D5 D400 A008 D5F8 00D4;
0550 0100 0138 1A0A FB0D 3274 D400 B49A B78A;

0560 A7D4 057B 3274 1B0B FB0D 3A61 D402 202C;
0570 0D32 54D5 F801 D401 0001 D59B B88B A807;
0580 FB0D 328F 07FB 2C32 8F47 5248 F332 7FD5;
0590 F800 A94A D403 0032 932A 2AD5 DF85 AE85;
05A0 F85F D400 BB3A AA4B 32A7 B9FA 0152 99FE;
05B0 3BB4 FC01 F3D4 0100 5FB7 D406 003A C7D4;
05C0 00E8 99D4 0100 5AD5 D406 003A D189 D400;
05D0 6AD5 F800 A797 32E9 FA01 32E1 8752 89F4;
05E0 A789 FEA9 97F6 B730 D587 A9D5 0C75 75C5;
05F0 BE3D AC19 3853 3E9B 4EC8 42E1 0265 A241;

0600 F800 B7A7 381A 0AFB 2032 05FB 2D32 3EFB;
0610 3132 3EFB 0132 3EFB 0332 3EFB 1532 05FB;
0620 0632 43FB 0732 48FB 0532 4D0A D401 D832;
0630 520A D403 0032 59D4 00A0 09F8 01D5 97A9;
0640 F800 D5F8 01A7 3005 F802 A730 05F8 03A7;
0650 3005 0AD4 00BB A930 5CD4 0590 8732 6A27;
0660 8732 7127 8732 7827 307F 9752 89F4 B730;
0670 0597 5289 F5B7 3005 D405 D289 B730 05D4;
0680 00E8 89B7 3005 9BA8 6845 818E EC3D DADC;
0690 381A 0AFB 2032 91D4 01D8 3AB0 4AB7 D402;

06A0 203D 0D3A B51A D406 003A AF97 D405 03D5;
06B0 D400 A00A D5D4 00A0 0BD5 D406 003A EDF8;
06C0 00A7 0AFB 0D32 F7FB 3132 EEFB 0232 F11A;
06D0 89B7 D406 003A ED97 5289 F532 E03B F427;
06E0 8732 E7F8 0030 E9F8 01D4 0100 01D5 2730;
06F0 CF17 30CF 1730 E089 32E7 30E3 E9C1 A4A3;
0700 D406 003A 2689 B7D4 005F D402 2025 033A;
0710 271A 0AD4 0300 3A2C D405 9097 5289 F33A;
0720 0AD4 0478 F800 D5D4 00A0 0CD5 D400 A00D;
0730 D5
```

Listing for 1802 PILOT. The listing is in UT4 format. The first four digits on each line are the starting address for the following data.

able to provide varied responses to repetitive situations.

The CONTROL instruction is provided as a convenience for such tasks as clearing the display screen or whatever; the END instruction is used to stop execution of the program.

Although 1802 PILOT does not use line numbers, any statement can be labeled with a number of range 0 to 255. The use of numbers instead of text names for statement labels permits program branching on the basis of computations performed in the JUMP and USE instructions.

All variables are identified with the single letters A through Z. Thus, there are 26 numeric and 26 string variables. In the TYPE and ASK instructions (where both types may appear), the variable type is identified with the $ or # signs; in all other statements only the numeric variable is used, and the # is omitted.

The Text Editor

An integral part of the 1802 PILOT interpreter is the text editor, which is used to create or modify programs. Table 2 lists the editor commands and their syntax.

The editor functions by storing ASCII character codes in a section of memory called the text buffer. The text buffer is variable length and expands or shrinks as material is added or deleted. The editor keeps track of where it is working by maintaining a text pointer (register RA), which always points to the start of the current line. For the most part, the user is burdened with the task of mentally keeping track of the text pointer.

The commands UP, DOWN, BEGIN and END move the text pointer. BEGIN and END move the text pointer to the start and end of the text buffer. DOWN and UP move the text pointer a specified number of lines (1 to 255) relative to the current position of the text pointer.

The INSERT command is used to add one line of text to the buffer. Each line of text to be inserted must start with the

1	*Division by zero*
2	*Cannot identify command*
3	*Memory full*
4	*Bad numeric variable syntax in TYPE statement*
5	*Cannot find text variable in TYPE statement (or bad syntax)*
6	*Bad numeric variable syntax in ASK statement*
7	*Bad text variable syntax in ASK statement*
8	*Undefined RETURN statement*
9	*Bad expression syntax*
10	*No variable name in COMPUTE statement*
11	*No "equals" sign in COMPUTE statement*
12	*Cannot find statement label of JUMP or USE*
13	*Bad statement label syntax*

Table 3. Error codes

Address	Present Code	Function
006F	813E	Address of routine to input one character from the terminal.
006B	81A4	Address of routine to output one character to the terminal.
0293	8140	Address of routine to input one character from cassette.
0275	81A4	Address of routine to output one character to cassette.
01C3	8000	Entry address of system monitor.
0244 028F	20 20	Upper limit of RAM available to PILOT—Number represents last page plus one—20 limits PILOT to 8K.
0125 0138	3E 3E	Sets maximum length of input line buffer.
0145	13	ASCII character code to move cursor left one space on TVT.
0146 014F	15 15	ASCII character code to erase to end of line on TVT.
0000	N/A	Cold and warm start address
0731	N/A	First byte of spare memory
07FF	N/A	Last byte of spare memory

Table 4. Key program memory locations.

letter I. The text pointer moves as text is inserted, allowing sequential entry of lines. The command opposite of insert is KILL. This command deletes the specified number of lines from the text pointer.

The WRITE command allows the user to review text material by typing out the specified number of lines (starting with the current line). This command is also useful when the user is uncertain of the location of the text pointer. This command does not move the text pointer or alter the text material.

The CLEAR command serves to initialize the text buffer. When the editor is first invoked, this command *must* be used prior to all other commands. The exception to this rule is when the editor is used on text already existing in memory.

The SAVE command dumps the entire contents of the text buffer to the cassette recorder (or other storage device). Similarly, the LOAD command fills the text buffer from the cassette. If text already exists within the buffer, the LOAD command will append the new text to the end of the existing text. This feature can be used to combine multiple text segments.

The final commands are MONITOR, which initiates a return to the system monitor, and PILOT, which initiates execution of the program.

Using 1802 PILOT

Operating the PILOT interpreter is a relatively simple process. After PILOT is loaded into memory, it may be invoked by the system monitor by loading a register (any register is permissible) with 0000H and making that register the program counter. For the UT4 monitor, use the command $P0. PILOT will respond by typing EDIT, followed by a prompt (right arrow).

The EDIT commands listed in Table 2 can now be used to either create a new program or to load an existing program from a cassette. Once the program is ready, the PILOT command is used to start execution. At the conclusion of the program the interpreter will automatically return to the EDIT mode.

For the convenience of the operator, the EDIT and PILOT modes receive operator input from a line buffer routine that allows for the correction of errors. Back space deletes the previous character from the input line buffer and echoes characters to the TVT to cause erasure of the character from the screen. Cancel (Control X) deletes the entire line from the buffer and screen.

As an additional aid to the operator, the line buffer routine accepts HTAB commands. A tab is assumed at every fourth character. This feature can be used to format the PILOT programs for improved readability.

If the interpreter detects a program error, it will immediately stop execution of the

```
HI/LO GAME FOR 1802 PILOT

    T:HI, MY NAME IS HI/LO.  WHAT'S YOUR NAME:
    A:$N
    T:
    T:I'M THINKING OF A NUMBER BETWEEN 0 AND 100.
    T:WHEN YOU ARE READY $N, TRY TO GUESS THE NUMBER.
    T:I WILL TELL YOU IF YOUR GUESS IS TOO HIGH OR TOO LOW.
INITIALIZE *1  C:N=0
               Z:101
START      *2  T:
               T:YOUR GUESS:
               A:#G
               C:N=N+1
               X:G=Z
               JY:3
               X:G>Z
               TY:TOO HIGH
               TN:TOO LOW
               J:2
CORRECT    *3  T:THAT'S IT $N! YOU TOOK #N GUESSES.
               T:
               T:WOULD YOU LIKE TO PLAY AGAIN:
               A:
               M:YES,SURE,OK
               JY:1
               T:BYE $N
               E:
>
```

Sample program. Note that although 1802 PILOT has no specific command for remarks, they can be included at will on separate lines or may precede the commands on statement lines.

```
HI, MY NAME IS HI/LO.  WHAT'S YOUR NAME?BOB

I'M THINKING OF A NUMBER BETWEEN 0 AND 100.
WHEN YOU ARE READY BOB, TRY TO GUESS THE NUMBER.
I WILL TELL YOU IF YOUR GUESS IS TOO HIGH OR TOO LOW.

YOUR GUESS?50
TOO LOW

YOUR GUESS?75
TOO LOW

YOUR GUESS?85
TOO LOW

YOUR GUESS?90
TOO LOW

YOUR GUESS?95
TOO LOW

YOUR GUESS?97
TOO LOW

YOUR GUESS?99
TOO HIGH

YOUR GUESS?98
THAT'S IT BOB! YOU TOOK 8 GUESSES.

WOULD YOU LIKE TO PLAY AGAIN?NO
BYE BOB

EDIT
>
```

Sample run of the HI/LO program.

Register	Assignment	
R0, R1	Not Used	
R2	Stack Pointer	Standard call and return except D is preserved through call and return by storing in RF.1 and R9 is saved/restored through calls (R2 through R6)
R3	Program Counter	
R4	CALL Program Counter	
R5	RTN Program Counter	
R6	Holds Return Address of Address of in-line argument	
R7	Save/Restore Register	
R8	Scratch Pad	
R9	General Purpose Counter	
RA	Text Pointer	
RB	Line Pointer	
RC, RD, RE	Reserved for I/O	
RF	RF.1 holds I/O ASCII character and D register	

Table 5. Register assignments.

Address	Function
0800	Error Code
0801	Match Flag
0802	Subroutine Count
0810	Text Pointer (Main)
0812	Text Pointer—Subcount = 1
.	. . .
083E	Text Pointer—Subcount = 24
0841	Numeric Variable A
.	. . .
.	. . .
085A	Numeric Variable Z
085F	Random Number Seed
0860	Start of Line Buffer
08FF	Top most byte of stack
0900	Start of Text (STX)
0901	1st character of text

Text Format:
STX First Line CR - - - - Last Line CR ETX
String Variable Format:
(Text ETX) $n contents - - - - $n contents ETX
NOTE: Strings are stored after text in memory. Each time a string variable is updated the old contents are deleted and the new contents are added to the end of the string file.

Table 6. Data storage format.

program and return to the EDIT mode. The interpreter will provide an error code as listed in Table 3. The program line containing the error will also be displayed.

If monitors other than UT4 are used, it will be necessary to patch the I/O routines to PILOT. Table 4 lists the codes that will require change. Replace the existing code with the absolute address of the appropriate I/O routine. The I/O routines must follow these conventions:

1. PILOT will echo characters input from the terminal back to the terminal. Therefore, the terminal and terminal input routine must not perform this function.

2. The I/O character is passed in D (and RF.1).

3. I/O routines must return to PILOT with an SEP R5 instruction.

4. Table 5 lists the PILOT register assignments. R2 through R6 are assigned according to a modified RCA standard call and return technique and may be used by the I/O routines to call subroutines. Registers R0, R1, R7 and RC through RF may be used for the I/O routines. If the I/O routines alter registers R8 to RB, they should be saved and restored.

Memory addresses 0731H to 07FFH are not used by the interpreter and are available if necessary for required I/O patches.

Also listed in Table 4 are the character codes that PILOT echoes to the terminal in response to the back-space and cancel commands. These codes should be changed as required to achieve the indicated functions.

Conclusion

1802 PILOT is usable on small systems with as little as 4K of memory. Since the interpreter takes only 2K of memory, this leaves 2K for programs, which is adequate for many purposes. Additionally, some means of displaying ASCII characters, such as a TVT, is required.

Although it takes several hours to hand-load the code in the PILOT listing, you will find the effort well worthwhile. Once you have 1802 PILOT on tape, you will enjoy writing and using the large variety of interesting programs that are possible with this language.■

John Warren
Dept. of English
E. Carolina University
Greenville NC 27834

Red-Handed Credit Grabber

Thwart credit-copping creeps with the tricks presented here.

One of the most despicable characters in the world of microcomputers is the Red-Handed Credit Grabber. His first act on buying a program is to delete the programmer's name and substitute his own.

With a little imagination you can impede his drive to self-aggrandizement—or at least make him work a bit harder for his ill-gotten gains.

It is axiomatic that the Grabber is a poor programmer—if he wasn't, he'd be writing his own programs instead of stealing yours. In general, his method of operation is to scan the program lines, locate the PRINT statement with the programmer's name and change it. If he can't find it, he can't change it.

CHR$ is a statement that converts a decimal number in ASCII code into its equivalent character. For instance, 79 converts to O, 65 to A and so on. You can use this statement to convert a series of numbers into a series of letters. With a FOR-NEXT loop, a READ from DATA and CHR$, you can have your credit line without ever spelling it out in the program.

First, convert the credit line into ASCII code. This can be done by consulting a table of ASCII equivalents or by running Program A. If a table is used, don't forget to include the spaces between words. Thirty two is the ASCII code for a blank.

Enter the ASCII codes as a dataline or as a series of datalines in your program. Then construct a FOR-NEXT loop to read the data and display the corresponding letter on the screen (Program B).

This approach is still vulnerable to the Grabber, as he may delete lines until he manages to eradicate the credit line. However, there are tricks that can still slow him down until he gives up in disgust.

Use a series of GOTOs and GOSUBs to confuse the track through the program. This is particularly effective where the initialization is lengthy or complex.

Instead of combining the READ and PRINT statements in one FOR-NEXT loop, use two loops. In one, load an array from data, and in the other, print the array through a CHR$ statement. These loops can be separated by other statements to confuse matters. (See Program C.)

To raise the frustration level of the Grabber to the boiling point, use the first two or three elements of the dataline to hold initialization variables for use in the program. For instance, I use the first three data arguments in my Space Lander program to hold the gravitational accelerations for the moon, Earth and Mars. This way the Grabber can remove the dataline or the READ statement and eliminate the credit line—but the program won't run.

Since the Grabber usually removes entire lines, you can make his job much harder by including initialization statements on each line that has to do with your credit line. Again, a deletion of the line will prevent the program from running.

The Red-Handed Credit Grabber is one of the most contemptible characters in the computer world, but with imagination and a few tricks, you can have him pulling out his own hair instead of your credit line. ■

```
10 CLS
20 PRINT"INPUT STRING TO BE DECODED"
30 INPUT A$
40 A = LEN(A$)
50 FOR B = 1 TO A
60 B$ = MID$(A$,B,1)
70 PRINT ASC(B$),
80 NEXT
90 PRINT "THE STRING IS ";A;" CHARACTERS LONG."
100 END
```

Program A—extracts ASCII codes.

```
10 FOR B = 1TO14
20 READ A
30 PRINT CHR$(A);
40 NEXT
50 END
60 DATA 66,89,32,74,79,72,78,32,87,65,82,82,69,78
NOTE: The number 14 in line 10 is the length of the message in letters.
```

Program B—prints credit line.

```
10 DIM A(17)
............
20 FOR X = 1 TO 17
30 READ A(X)
40 NEXT
............
50 FOR X = 4 TO 17
60 PRINT CHR$(A(X));
70 NEXT
............
1000 DATA 980,162,372,66,89,32,74,79,72,78,32,87,65,82,82,69,78
```

A(1), A(2), A(3) are the gravitational accelerations of the Earth, the moon and Mars, respectively. They are used later in the program. The dotted lines indicate where the various parts of the program can be separated by GOTOs, GOSUBs or program lines. The number 17 that appears in lines 10, 20 and 50 is the length of the message plus the number of variables in the dataline.

Program C. Smokescreen program to print credit line.

ENTER A WHOLE NEW WORLD with KILOBAUD MICROCOMPUTING

. . . and get your first issue FREE!

Hardly a day passes without the computer having some effect on millions around the world. The world of microcomputing is changing every day, and there's one way you can keep up to speed in this dynamic field: Subscribe to Kilobaud **MICROCOMPUTING. MICROCOMPUTING** magazine is the forerunner in small business and home computing, loaded with articles on the most popular microcomputers available: the TRS-80, Apple, PET, Heathkit, Imsai and Altair, to name a few. In addition, the articles cover a whole range of subjects, from business applications to educational uses, from product reviews to the latest in software.

As a special limited offer to new subscribers, we'll send you your first issue absolutely free! You'll receive 13 issues for the price of 12. And we're offering you a year's subscription at half the newsstand price . . . only $15 (it's $30 a year on the newsstand), and the only difference between a subscriber copy and a copy of **MICROCOMPUTING** on the newsstand is $1.25. Send in the card today and we'll deliver your first copy in 4-6 weeks.

ENTER A WHOLE NEW WORLD
WITH KILOBAUD MICROCOMPUTING
And Get Your First Issue Free!

☐ Send me my first issue of **MICROCOMPUTING** free and start my subscription into the new world of **MICROCOMPUTING.**
☐ One year (12 issues) at half the cover price—$15.
☐ I prefer the additional savings of 3 years (36 issues) for $45. I understand that I'll still receive my first issue free.
☐ Payment Enclosed ☐ Bill Me Later
Bill my ☐ Master Charge ☐ BankAmericard ☐ Visa

Card #______________ Exp. Date ______________
Signature______________________________
Name ______________________________
Address ______________________________
City ______________ State ________ Zip ________

Offer good in U.S.A. and possessions only. Canadian subscriptions add $2.00 per year. All foreign, one year only—$23 payable in U.S. currency only.
MICROCOMPUTING Subscription Department ●
P.O. Box 997 ● Farmingdale NY 11737 197

ENTER A WHOLE NEW WORLD
WITH KILOBAUD MICROCOMPUTING
And Get Your First Issue Free!

☐ Send me my first issue of **MICROCOMPUTING** free and start my subscription into the new world of **MICROCOMPUTING.**
☐ One year (12 issues) at half the cover price—$15.
☐ I prefer the additional savings of 3 years (36 issues) for $45. I understand that I'll still receive my first issue free.
☐ Payment Enclosed ☐ Bill Me Later
Bill my ☐ Master Charge ☐ BankAmericard ☐ Visa

Card #______________ Exp. Date ______________
Signature______________________________
Name ______________________________
Address ______________________________
City ______________ State ________ Zip ________

Offer good in U.S.A. and possessions only. Canadian subscriptions add $2.00 per year. All foreign, one year only—$23 payable in U.S. currency only.
MICROCOMPUTING Subscription Department ●
P.O. Box 997 ● Farmingdale NY 11737 197

Troubleshooting Tips and Techniques

A case is presented here for having a front panel on your system before it breaks down.

Bob Bosen
Box 93
Magna UT 84044

Sooner or later all computers grow hair and howl at the moon. I can make that statement with a great deal of confidence because I've spent the last two years flying all over the country repairing computer systems that were built correctly, maintained properly, designed conservatively, but still had occasional failure due to Murphy's Law of Computing Machines, which clearly states: "Sometimes it does not compute!" Four Murphy Axioms follow: (1) no computer is big enough; (2) no computer is fast enough; (3) no computer has enough memory; (4) no computer has enough peripherals.

Computers, by their very nature, are anxious to grow. A normal computer system will always expand until one of the following two conditions is met: (1) expansion exceeds available finances; (2) expansion exceeds design's ability to keep everything under control.

A well-designed system should spend a lot of time under condition 1 until condition 2 is reached, but eventually condition 2 catches up with virtually every computer. At this point (if not sooner), hardware problems develop.

This article is the result of the current trend to eliminate the traditional front panel from hobbyist computer systems in an attempt to save the cost of a fistful of LEDs, ICs and switches. I will discuss some reasons for a front panel, the various types available on hobbyist systems and some system troubleshooting techniques utilizing a binary front panel and a minimum of test equipment.

In The Beginning . . .

First of all, let's address the question of why front panels loaded with binary lights and switches ever became prevalent on computers.

Anyone who has ever toggled in a monitor program or lengthy bootstrap knows that they were not put there for programming! No serious designer could ever have believed that the toggle switches typically used could stand up to the hundreds of thousands of flips and flops that would be necessary even if programmers' hands evolved to withstand the pain and suffering inflicted. The truth is that terminals are for programming and front panels are for hardware debugging.

We have already established that computers are complicated machines with a tendency to outgrow their own abilities to stay under control.

Now the reasons behind the traditional front panel should be clear. Computer designers haven't been trying to eliminate programming terminals; they have been facing facts. The traditional front panel is the result of the economic fact that it is cheaper in the long run to spend a few extra dollars on lights and switches than to to try to repair a 16- or 18-bit high-speed computer or data channel with a dual-trace scope and a good-luck charm!

Types Available

With these facts in mind, then, let's examine the various types of front panels available on hobbyist computers. They fit into about five general types.

1. Simple lights and switches for control of memory.
2. Switches and logic to control the processor and, in turn, the memory.
3. Hex or octal keypads and displays for control of memory.
4. ROM programs that accept data from keypads and copy data into displays.
5. ROM monitors that accept data from a terminal and copy data onto a CRT.

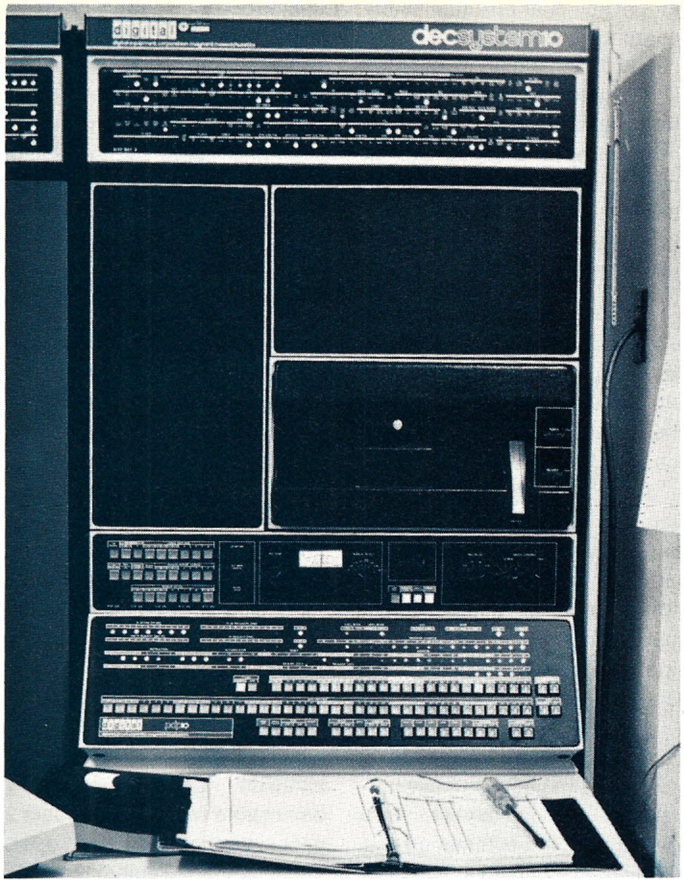
The last type is the easiest to implement in hardware because it just requires a socket for the ROM and an interface to a terminal that will have to be built anyway. These factors have combined to make type 5 very popular among small, inexpensive computers.

My description of a type 4 system was purposely worded to sound like type 5 because I wanted to illustrate that the intent is the same: to build a programmer's terminal. Neither system is effective at hardware debugging because both depend on every major component in the system. If anything goes wrong, nothing works at all, and the debug panel is as useless as the bad component responsible for the problem in the first place.

In fairness I should point out that there are certain types of problems that might not be fatal to ROM-based debugging, but a lot of things *are* fatal! The advantage of the type 4 system over type 5 is that if the terminal dies, the debug panel can still be used, and programs may be written to help debug the terminal.

Panels of types 3 and 4 look a lot alike from the outside world, but the superiority of type 3 is so important that serious investigation is in order when you see a photograph that could represent either type. It is possible to design a front panel with hex or octal keypads and readouts that can fully control a memory system without dependence on the microprocessor. This can be an effective troubleshooting aid when the system is down, and is slightly effective for limited programming (bootstraps, etc.).

Another advantage of this method over the ROM versions is that no computer address space is committed to the front panel ROM, allowing flexibility in memory allocation. This is an important consideration because when you are configuring your system you probably don't know what software will become available to you in the future or where it will reside. It is a major job to relocate software around spaces that are unnecessarily tied up to sup-



Here's the business end of a typical large computer. Notice the hardware tools: a screwdriver, a system logbook and a binary front panel.

port ROMs. So, consider these factors when you are selecting a computer system.

The Altair 8800A is a type 2 system providing front-panel logic to jam instructions into the processor in order to do the desired work at the request of the operator through the front-panel switches. This system works and has some independence with problems on the 100-pin bus because the signal path between the front panel and microprocessor is separate. This is useful for finding shorts or broken wires on the bus (more on this later).

The type 1 system is easy to implement in hardware, and, for that reason, could be used in an inexpensive computer system without significantly increasing the cost. Its principle of operation would be to take over the memory bus using the Tri-state or open collector logic any device would use for a DMA transfer, then to just hook each address and data line to a switch and a light so it could be controlled and monitored.

This is not beyond the scope of most hobbyists, and could be built into a PC board and plugged into most computer backplanes, providing effective control of the system memory bus at a minimal cost. Systems that already use this scheme can sometimes be identified by their low cost and lack of a "deposit next" switch, although this is by no means an ironclad rule.

Maintenance Approach

I hope by now you are beginning to be convinced of the importance of lights and switches as a hardware debugging tool. Using lights and switches, I have solved problems in minutes that would have taken days (if not weeks) with just a scope.

I would like to talk about some of the tricks that can be used to isolate problems in a typical hobbyist computer with a front panel and a minimum of tools. To a large extent, these tricks depend on advanced preparation, which can be the secret to keeping your system running with hardly an interruption.

Accept the fact right now that sooner or later your system is going to quit working. At that point you may be completely lost, or you may be able to follow a simple, methodical strategy to find and fix the problem. The choice is up to you, but it will be determined by what you do when your system is running, not by what you do after it dies.

A good way to begin preparing for the death of one of your peripherals is to note—in great detail—what each part of your system does. Punch in a few simple programs, and, when everything is running normally, make a note of all of the lights on your front panel. Take notes with sufficient detail so that after two hours of pulling your hair out you will be able to refer to the notes and know for sure which lights should be on and which should be dim or off during normal execution of three or four different, simple routines.

Then begin pulling boards out of your backplane one by one. Jot down what you can and can't do as each is removed. Write programs that exercise each specific board, and make a list of how many other boards must be working perfectly for each program to run. Pay particular attention to the absolute minimum configuration in your system that can support a simple program, like a jump to self loop. This will usually require just a CPU card and a RAM card, but some CPUs contain enough RAM to run a program like this all by themselves. Try it.

Make a note of the lights and switches, realizing that you may have to go to this simplest configuration someday to find out if anything works at all; in that situation you won't be willing to trust anything. This kind of exercise can entertain you for several evenings, but after you've done it, you will have learned a lot about the way the various modules of your system interact, and you'll be well prepared for the inevitable crash when it comes.

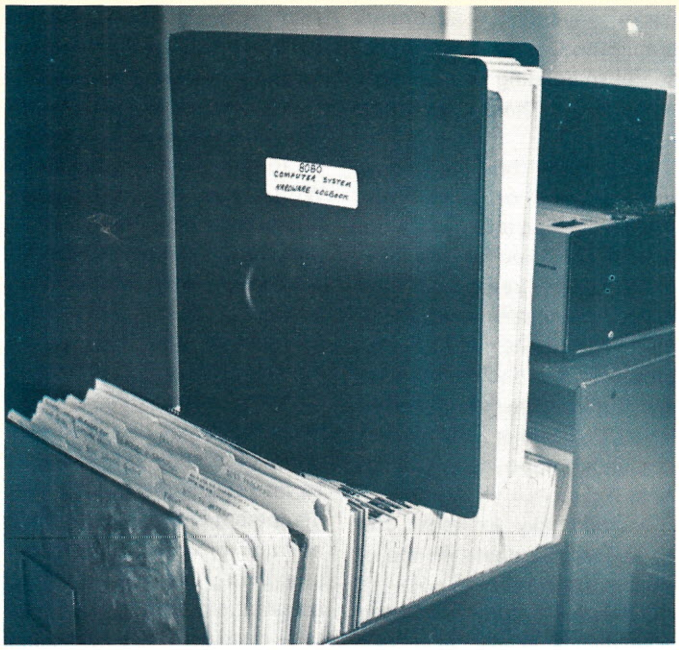
Another area where advanced preparation is essential is the system's paperwork. When you begin troubleshooting you will need readable schematics and block diagrams of *everything* on your system. Save all the schematics from the various manufacturers in a system logbook, and make detailed descriptions of any modifications or equipment you add. Never let anyone modify your equipment or make any kind of addition or correction without providing you a detailed schematic of everything done.

Organize this logbook chronologically and use it to keep track of system troubles. Every time your computer does something that might be a hardware problem, write it down. Sometimes a pattern will emerge to reveal the cause of the problems. This may take months, so don't trust your recollection to decipher these things—write them down!

It has often been found that temperature or power-line fluctuations were the cause of problems previously blamed on other factors. I am familiar with one situation in which the failure of a computer in a hospital was due to the operation of a centrifuge in a nearby laboratory. The problem was found by comparing the failure log kept by the computer with the activity log of the laboratory personnel!

While your system is running you can also use whatever test equipment you have to explore normal system behavior. Dig out that old oscilloscope if you have one and hook it up to your system clocks, data and address buses to see what those signals look like. Get accustomed to the ugly spikes on the signals, so when you are troubleshooting real problems you won't be misled by something that is normal for your system. At this point you might get a few surprises, too!

You may find that your scope probes are shorted out and your system won't run at all



A second, indispensable hardware maintenance tool is a thorough, accurate system logbook. Keep it in a safe place next to your back issues of Kilobaud.

with the scope hooked up! Imagine the frustration you are saving yourself by finding these things out on a working system! There may be a few places in your system where hooking up a scope or meter will cause enough noise or loading to create a failure, especially if you are using faulty or bargain-basement equipment. Make notes of these things, along with sketches of data bus signals during execution of some simple programs.

Once you do the things I've outlined above and develop the habit of keeping an accurate system log, you will have taken a giant step toward keeping your system running smoothly, at a minimum cost. Now enjoy its operation till the day it decides to die; then you'll be able to pat yourself on the back for the extra effort you've put in.

Troubleshooting Tricks

When your system does quit on you, your basic troubleshooting procedure will be to first eliminate all unnecessary complications by removing all peripherals and circuit boards that are not absolutely necessary for operation of the system to exercise the problem. Eliminate things one at a time, and test the problem after each part is removed to see if it is responsible for the failure. Many computer modules are capable of creating failures in a seemingly unrelated part.

Next, compare the faulty system behavior with the correct behavior you have described in your system log. Look for the big picture, generalizing the symptoms and looking for the biggest ones first, always trying to find a single pattern that will explain all of the symptoms and relate them to the failure of a single signal or component. This method will usually lead you to a pretty good suspicion of which board or module has caused the problem.

Then, take a look at the spare boards you have liberated from the system in the simplification process described above, and see if one of them can be made to perform the function of the board in question. Memory boards can, for example, almost always replace each other. If you can successfully replace the faulty board and solve the problem, you will have taken a big step toward repair, and, more important, you can probably get your system running again even if you have to get by without the faulty board for a while.

The front panel with lights and switches will start to pay for itself as you begin looking at the symptoms of your broken system and compare them with correct records. Some of the things you should be able to spot will be:

1. Address and data lines shorted high or low. These will be easy to spot because no matter what you do, the corresponding light will be stuck on or off.

2. Address and data lines shorted to each other. These will be indicated by a tendency of both lines to be pulled low whenever either of them should be low, with both going high together, but only when both are supposed to be high.

3. Control signals faulty. There should be indicators to tell you if the bus is under control of the processor, front panel, or some external device. Watch that these are not shorted to some other line, or high or low, as described by the previous two paragraphs.

4. Processor status wrong. There should be indicators to inform you of the arrival of an interrupt, stack operations, or the types of instruction or operation being performed. If these don't match your records, they are probably the source of your problem.

You can do a lot with the switches and lights to verify your suspicions from the above indicators. For example, try to turn on each of the lights one at a time by examining addresses that use the corresponding address lights, and by depositing a bit in memory, then moving that bit one position to the left and depositing it in the next cell and so on until you've stored a bit in each possible position. Read them back to make sure they actually went where you put them.

This kind of exercise will usually locate shorted backplane runs, unsoldered pins, and sometimes even faulty chips. Remember that shorted pins tend to pull each other low while open wires tend to pull high. Often a visual in-

BINARY		OCTAL		HEXADECIMAL	
Correct	*Wrong*	*Correct*	*Wrong*	*Correct*	*Wrong*
0000	0000	00	00	0	0
0001	0001	01	01	1	1
0010	0000	02	00	2	0
0011	0001	03	01	3	1
0100	0000	04	00	4	0
0101	0001	05	01	5	1
0110	0110	06	06	6	6
0111	0111	07	07	7	7
1000	1000	10	10	8	8
1001	1001	11	11	9	9
1010	1000	12	10	A	8
1011	1001	13	11	B	9
1100	1000	14	10	C	8
1101	1001	15	11	D	9
1110	1110	16	16	E	E
1111	1111	17	17	F	F

Table 1. The columns of this table represent four bit patterns stored in a faulty memory. The two middle bits are shorted together in this example. For each of the three numbering systems, the "correct" column represents what should be in this memory nibble, and the "wrong" column represents what is actually read out. Notice how obvious the short is in the binary lists: Whenever either of the two center bits is low they are both pulled low. In hex or octal, however, the pattern is much less obvious. This is a very common type of problem in a computer system because these data lines connect with hundreds of different chips, each containing thousands of gates. Notice further that a ROM-based front panel would not work at all with this kind of a fault.

spection of the areas you suspect after these exercises will reveal one of these conditions.

At this point I would like to mention the advantages of a binary front panel over an octal or hexadecimal system. While it is true that binary is a hard language with which to do serious programming, I've never seen a hexadecimal short-circuit! Octal shorts are equally rare. The real beauty of a binary front panel is never more obvious than when you're looking for problems related to circuit failures, because these are almost always binary in nature. Binary front panels tend to be simpler, and when you have nothing to work with but a dead computer and a questionable front panel, you might as well give up. You need something you can trust completely before you can ever hope to repair a computer (Table 1).

Summary

One of my main objectives in writing this article was to effectively challenge computer module manufacturers to devote some of their talents to development of inexpensive, simple, reliable sets of lights and switches compatible with the popular hobbyist computer systems available. I'm sure a lot of hobbyists would pay $100 or so for a kit of parts that would provide a minimum of system monitoring and control for hardware debugging and limited programming.

The manufacturers have already proven their ability to give us cost-effective, state-of-the-art developments in mass-produced quantities. I'll bet they could come up with some really exciting ideas in this area. Let's see some breakpoint logic, power supply monitoring, protect violation indicators, parity error indicators, programmable LED registers, switch-controlled input ports, front-panel interrupts, master clear buttons, and your ideas, too!

In conclusion I will confess that I am biased in favor of binary front panels. (I'll bet you couldn't tell!) I recognize that there are other effective ways to troubleshoot computers, but I am convinced that lights and switches will almost always pay for themselves in a computer system that gets used and expects to grow.

I recommend that all computer owners and potential owners give consideration to the philosophies and facts I've expressed here as they consider the best way to turn their dollars into useful computing machinery. Best of luck to you all, and I hope you won't be needing to refer to this article frequently! ■

CAST A SPELL. WIN A SORCERER.

Super Starter Kit

This review of Technico's kit also contains food for thought about your computing power.

Richard L. Mataka
60-10 69th St.
Maspeth NY 11378

With the advent of the 16-bit microprocessor, a whole new world of computing power has opened up to personal computerists. No longer are you tied to the standard S-100 bus system but are allowed to let your creations—in either hardware or software—roam free.

Your computing power is immensely increased due to the machine's language and the speed at which it works. It contains features you may have thought about but have never seen. Read on and learn about Technico's "Super Starter Kit," which is based on the Texas Instruments TMS 9900 (see Fig. 1).

More Than an 8-Bit Machine

Unlike the conventional 8-bit microprocessors, the TMS 9900 has a 16-bit data word. Within this one word, most instructions are carried out. It tells, for example, what register you are going to or coming from, mode of addressing, distance of jumps and other such pertinent information. This is known as microprogramming, which is used extensively throughout the instruction set.

Within the TMS 9900 microprocessor there are only three hardware registers: the program counter, status register and workspace pointer register. The program counter functions the same as the PC in any other processor. The same almost holds true for the status register.

The exception to this is that bits 12 to 15 are used for an interrupt mask. There are 15 possible levels of interrupts that can occur in this system. This interrupt mask determines the lowest priority level that the processor will recognize.

Another factor that is different from the 8-bit machines is the absence of an accumulator. What you are given is a set of 16 scratchpad registers that reside in memory. These registers are referred to as workspace registers and go by the designation of R0 to R15. Most data manipulation is done through these workspace registers that reside in RAM memory.

One exceptionally nice feature is that you can redesignate your workspace registers to any area of memory by just changing the workspace pointer. There are also provisions for storing the workspace pointer address so that it is possible to have multilevel subroutines, each of which contains its own workspace registers.

It is through the microprogramming of the instructions that the workspace pointer register knows which workspace register in memory is going to be used. It is a simple concept that adds a great amount of flexibility in programming.

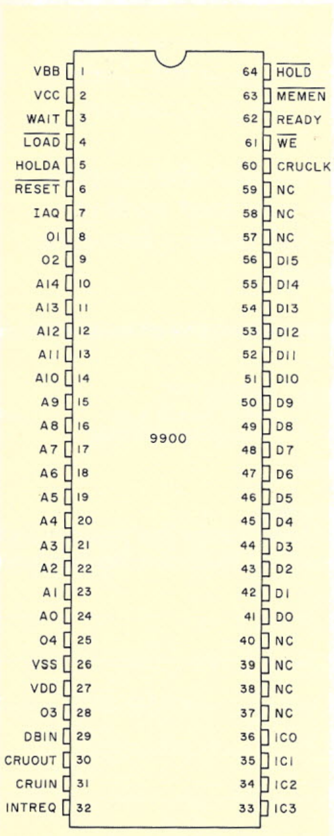


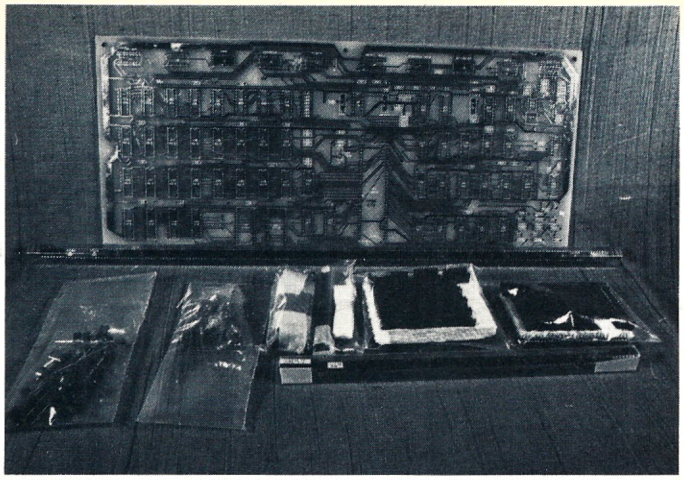
Fig. 1.

FORMAT (USE)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Arithmetic	OP CODE			B	TD		D				TS		S			
2. Jump	OP CODE								SIGNED DISPLACEMENT							
3. Logical	OP CODE						D				TS		S			
4. CRU	OP CODE						C				TS		S			
5. Shift	OP CODE								C				W			
6. Program	OP CODE										TS		S			
7. Control	OP CODE											NOT USED				
8. Immediate	OP CODE											NU	W			
	IMMEDIATE VALUE															
9. MPY, DIV, XOP	OP CODE						D				TS		S			

KEY

B = Byte Indicator (1 = Byte, 0 = Word)
TD = Destination Address Modification
D = Destination Address
TS = Source Address Modification
S = Source Address
C = Transfer or Shift length count
W = Workspace Register Number
NU = Not Used

Fig. 2. Instructions format.



The Super Starter Kit.

There are 16 bits used for addressing, but only 15 of them are brought out to pins on the chip. These are A0 to A14. This allows for direct addressing of 32K worth of memory. The sixteenth bit (A15) is used internally by the CPU for byte (8 bits)-oriented instructions. Even though you can only directly address 32K of memory, you must remember that 32K with this machine is equivalent to 64K in any 8-bit processor. This means you have a lot of memory for your programs.

There are six possible addressing modes from which you can choose. Many times it is possible to have more than one type of addressing mode within a single instruction. The six modes are inclusive of all combinations that can be utilized.

1. Immediate addressing mode—with this the operand specifies the register, and the word following this is the data that is to be put into that register.

2. Register addressing—here the operand is contained in the workspace register.

3. Register indirect—in this mode the data will be deposited in the memory area that is specified by a value in one of the workspace registers.

4. Register indirect with Auto Increment—the same as the indirect addressing mode, except upon completion of the instruction your register is incremented by one.

5. Indexed mode—in this mode you add a constant number to a specified workspace register to arrive at the memory location where you are going to put your data.

6. Relative mode of addressing—with this you are allowed to move your data to a location that is + or − 127 words from your present location. (Note: All information moved by addressing can be data or instructions.) As you can see, there is more sophistication here than is normally found in other processors.

An interesting fact in the design philosophy behind the TMS 9900 is that it is based on TI's 990 minicomputer. By this I mean that the instruction sets for the two are almost identical. Of the 72 instructions in the 990, the TMS 9900 recognizes 69 of them. In many cases this allows for direct compatibility between programs for the two systems. Under these circumstances, there already is a nice library of software available for the user.

Beware: The Super Starter Kit subtracts a few more instructions from that total number of 69. The ones that are subtracted are used for external controls and have no real effect upon this system's operation.

For all of the instructions there are a total of nine different formats (see Fig. 2). The manner in which the instruction is microprogrammed depends on the conditions that you set up within the various fields for that instruction.

The "B" field in Format 1 (Fig. 2) when set to a 1 indicates that you are going to do byte arithmetic. If it is a 0 then it is understood the whole word is

MNEMONIC	INSTRUCTION
A	Add (word)
AB	Add (byte)
ABS	Absolute
AI	Add immediate
ANDI	And immediate
B	Branch
BL	Branch and link
BLWP	Branch and load workspace pointer
C	Compare (word)
CB	Compare (byte)
CI	Compare immediate
CLR	Clear operand
COC	Compare ones corresponding
CZC	Compare zeros corresponding
DEC	Decrement (by one)
DECT	Decrement (by two)
DIV	Divide
IDLE	Computer idle
INC	Increment (by one)
INCT	Increment (by two)
INV	Invert (one's complement)
JEQ	Jump on equal
JGT	Jump on greater than
JH	Jump on high
JHE	Jump on high or equal
JL	Jump on low
JLE	Jump on low or equal
JLT	Jump on less than
JMP	Jump unconditional
JNC	Jump on no carry
JNE	Jump on not equal
JNO	Jump on no overflow
JOC	Jump on carry
JOP	Jump on odd parity
LDCR	Load CRU
LI	Load immediate
LIMI	Load immediate to interrupt mask
LWPI	Load immediate to workspace pointer
MOV	Move (word)
MOVB	Move (byte)
MPY	Multiply
NEG	Negate (two's complement)
ORI	Or immediate
RTWP	Return workspace pointer
S	Subtract (word)
SB	Subtract (byte)
SBO	Set CRU bit to one
SBZ	Set CRU bit to zero
SETO	Set ones
SLA	Shift left (zero fill)
SOC	Set ones corresponding (word)
SOCB	Set ones corresponding (byte)
SRA	Shift right (MSB extended)
SRC	Shift right circular
SRL	Shift right (leading zero fill)
STCR	Store from CRU
STST	Store status register
STWP	Store workspace pointer
SWPB	Swap bytes
SZC	Set zeros corresponding (word)
SZCB	Set zeros corresponding (byte)
TB	Test CRU bit
X	Execute
XOP	Extended operation
XOR	Exclusive OR

Fig. 3. 9900 instructions.

going to be used. The TD/TS fields indicate the various addressing modes that you can use between the destination (D) and source (S) locations (might be memory or registers). The "W" field designates an actual workspace register that you are going to refer to. When they are combined as they are in this instruction format, you have a capable machine with a powerful language.

Earlier, I mentioned that the Super Starter Kit did not recognize all 69 instructions of the TMS 9900. Before proceeding, I should clarify this: A total of 66 are recognizable with this kit. (See Fig. 3 for instructions used with the Super Starter Kit.) The instructions not used in no way affect the kit's operation.

The four unused instructions are all recognized as valid, but they do not perform any processing functions as they are for external operations. These are the CKOF (Clock Off), CKON (Clock On), LREX (Load ROM and Execute) and RSET (Reset). These four instructions are in no way useful to this system's operation.

The Kit

The Technico Super Starter Kit can almost be thought of as the "dream machine." The parts you're given and the design concepts behind them are definitely noteworthy. First of all, you receive a detailed, well-written manual full of useful documentation in a three-ring binder. It not only includes an excellent hardware assembly section but also much general information on the kit's operation. This serves to familiarize you with the overall concept of the Super Starter Kit.

Next, even though you are only given 256 words of RAM memory, you are supplied with the sockets for the full 1K possible on board. This also holds true for all the ICs on the card as they are all in sockets. The board's high quality can easily be seen upon examination.

Finally, a large plus factor for the Super Starter Kit is the incorporation of a 2708 PROM programmer right on the board. This PROM programming area serves a dual purpose. When it's not in use, you have the ability to run programs that were previously on a 2708 PROM. This distinction between programming or running a program from PROM is made with the flip of a switch.

Another excellent aspect of the Super Starter Kit is that you are not tied down to any bus system. It could be considered a universal bus in that you don't have to use all of the signals, but they are brought out just in case you want them. It is the manner in which the signals are brought out that is an interesting design feature. It is simple but effective.

STEP 1.	Parts verification
STEP 2.	Install sockets
STEP 3.	Install resistors
STEP 4.	Install potentiometer
STEP 5.	Install capacitors and inductor
STEP 6.	Processor power jumper
STEP 7.	Install diodes
STEP 8.	Install transistors
STEP 9.	Install ICs
STEP 10.	Install expansion memory ICs
STEP 11.	Memory configuration
STEP 12.	Input/output configuration
STEP 13.	Install control switches
STEP 14.	Short test
STEP 15.	Connection of power
STEP 16.	Power check
STEP 17.	Install processor
STEP 18.	Connection of terminal
STEP 19.	Start Monitor

Fig. 4. Assembly steps.

All of your I/O, control and power signals are brought out to ten 16-pin IC sockets labeled J1 to J10. From this point on all you need are ribbon cables, and the signals can go to any device you wish. Since you are not tied down to any specific bus structure, you could even incorporate the best aspects from various structures to create your own bus system.

Thus, the universal bus has no edge connector to inhibit your choice of peripheral devices. It is even possible to hook up some of the speciality cards available with other bus systems with some interface design and sharp programming on your part. Where the lines can lead you is left up to you, your initiative and imagination.

Assembly

Building the Super Starter Kit is fairly simple because of the fine documentation. It takes you step by step through all construction phases to the finished product. Besides the manual, you only need a minimum amount of tools to build the kit.

There are a total of 19 steps in the assembly section, which, I suggest, you should follow closely (see Fig. 4). This is especially true for the pre-power checks, because if a major error has occurred it could show up here. (For Super Starter Kit specifications, see Fig. 5.)

All of the steps in the assembly section are thoroughly explained. For those not familiar with resistor and capacitor coding, there are charts that explain them. If, when building the kit, you adhere strictly to all of the steps, you should have no trouble. With care and patience, you should be able to assemble it in 15 hours. The time and effort are well worth it when you begin programming on the finished product.

As I was building the kit I took extra care in the installation of the IC sockets. I bent every pin against the solder landing, being careful not to cause any shorts. For the discrete components, I did the same thing and then clipped them as close to the board as possible. (Don't throw away these clipped leads, as they will come in handy for the various straps to be made later.)

PC board 7" by 16"
Sockets 68 supplied
Possible 1K words RAM (256 supplied)
Possible 1K words EPROM
Possible 1K words ROM
TI 9900 microprocessor
8 vector interrupt
RS 232, RS232C or 20 mA loop interface
Instructions 69
Maximum clock rate 3 MHz

Fig. 5. Super Starter Kit specifications.

Through each installation step I first placed all components in their proper positions on the board. Once all of the components of one type were installed, I checked to see if there were any leftovers. After double-checking to ensure everything was properly placed, I then soldered them to the board using a low wattage soldering iron. Then I clipped the leads and placed them aside for later.

Before going on to the next step, I removed the excess flux with a solder aid with a stiff wire brush on one end and inspected my work. I immediately cleaned up anything that appeared as though it might be shorting.

By doing it this way, I was able to spot my errors before they caused any major damage. After all the components were mounted on the board, I used flux remover and again inspected all of my work.

This double- and triple-checking may seem like a waste of time, but it definitely pays off in the long run. As it happened, I did spot a few

OHIO SCIENTIFIC DOES IT AGAIN

Ohio Scientific has taken its standard C3 computer and married it to the new Shugart 29 Megabyte Winchester Drive. The result is the C3-C. This new microcomputer now fills the vacuum that existed for computer users who need more mass storage capability than floppies can offer – yet until now, could not justify the additional cost of a larger capacity hard disk computer such as our C3-B 74 Megabyte disk system.

Winchester Technology

Winchester hard disk drives offer small business and professional computer users the logical solution to mass storage problems that are beyond the capability of floppy disks. In addition, Winchester disks feature a track seek-time that is much better than floppies and because they spin at eight times the rate of floppies, Winchesters have a shorter latency. Both of these points reflect one remarkable speed advantage Winchester disks have over floppies.

Coupled to the C3 Computer

Ohio Scientific's award winning C3 computer is a classic. It is the only computer series that utilizes the three most popular microprocessors – 6502A, 68B00 and Z-80. This tremendous processor versatility enables one to utilize a seemingly endless selection of quality programs available from Ohio Scientific's software library as well as from many independent suppliers.

And Advanced Software

For instance, there are single user, multi-user and network operating systems. A complete turnkey small business package, OS-AMCAP provides accounts receivable, accounts payable, disbursements, cash receipts, general ledger, etc. OS-CP/M offers a complete FORTRAN and COBOL package. And there is WP-2, a complete word processing system. For information management, OS-DMS, features an advanced file handling system and program library that simplifies information storage and recall and routinely performs tasks which usually require special programming on other systems.

Yields the Microcomputer of the Future

With an eye toward the future, the C3-C, like all other C3's was designed with provisions for future generation 16 bit microprocessors via plug-in options. There are ten open slots for lots of I/O and multi-user operation. Truly, the Ohio Scientific C3-C is a computer with a future.

The new C3-C computer with 29 Megabyte Winchester Hard Disk.

OHIO SCIENTIFIC

1333 S. CHILLICOTHE RD., AURORA, OHIO 44202 (216) 562-3101

OEM pricing available

shorts that could have caused major trouble. This was because in some areas the etching was very close. Use extreme caution.

The troubleshooting techniques in the manual are excellent and should be followed to the letter before the processor chip is installed. They entail "Ohming out" the power pins to assure there are no shorts. After all, it is always possible to overlook one. Also, if power is applied to the board with some of the chips installed, it could cause major damage to the chips, which results in a costly setback.

Once you are sure there are no power shorts, you can hook up your power supply. The board utilizes four voltages, one of which is optional. The voltages that are needed for the system's operation are +5 V @ 5 Amps, +12 V @ 3 Amps and −5 V @ 2 Amps. If you plan to program PROMs with your unit, you must also supply a +28 V @ 40 milliamps.

The current ratings given with the voltages are for the Super Starter Kit, so it is possible to get away with less. Technico also sells power supplies for this system in a modular form so that you only need to buy the voltages that you want to use. (See Fig. 6 for supply costs.)

There is still another check that you must perform before you plug in your processor chip. This has to do with checking the supply voltages that you hooked up at the respective pins on the ICs. You should do this cautiously, applying power for only a few seconds at a time so you can just get a reading and be assured that everything is operating properly. If all of these checks are followed, the installation of the processor chip, the connection of your terminal and the subsequent power-up should give you an operating microcomputer.

Be sure that while you are building the kit you are aware of the modification sheets in your manual. At present, there is only one, which has to do with the addition of a capacitor across the reset pin of the processor chip. Originally, when you tried to reset your unit it would get hung up and your Monitor would not respond properly.

Another factor requiring caution is the installation of the memory and I/O configuration straps. If you are going to use the standard configuration employed with the Super Starter Kit, the straps should be installed as shown. This is where the resistor leads come in handy as you can use these for the straps that have to be made.

All necessary voltages to run the Super Starter Kit retail for $139. The extra 12 volt supply that is needed to run a full system of 32K requires the expenditure of an additional $59.

Fig. 6. Power supply costs.

Operation

Knowing that the memory and I/O can be changed gives additional flexibility to this outstanding unit. It is possible to change the location of your Monitor to an entirely different memory location. This is if you are using this unit for your own application and not as the standard kit.

After you have gone through and painstakingly constructed the kit, the next step is to use it. Once you have powered it up and hit the reset, you should hit a carriage return on your printer. At this point, the Super Starter Kit should respond with ? as a prompt signal. If it doesn't, hit the reset a few more times and a carriage return again.

If you still have no prompt, check your terminal's wiring with the kit to be sure that all is correct. There is no need to set up baud speed with the Super Starter Kit; all of this is taken care of in the Monitor. Most likely what you will find is that you have a wire crossed between your terminal and the kit.

If that is not the case, inspect all of your work again, making sure that all ICs are in their proper locations and transistors and diodes are in the correct ways, check your strapping options, reexamine the board for shorts and try it again. Now you will probably find that it works all right. Now begins the fun part, the process of learning to program.

Monitor

The full name of the Monitor used with this kit is the "Mighty Monitor," and it is appropriately named. With this Monitor you are given 12 different commands from which you control all actions of your processing unit (see Fig. 7).

The commands I have used mostly are Alter, Breakpoint, Copy Memory to Memory, Dump, Hex Arithmetic, Go and Program EPROM. Workspace Dump, Snap, Inspect and Modify CRU (Communications Register Unit) bit are mostly for debugging programs. Load Program from terminal is used if you are loading a paper tape from your terminal.

What follows are short explanations on how the commonly used commands are utilized from your terminal.

A—Alter the contents of RAM. This command is used when you want to change a specific location in memory. The manner in which it is performed is to type an "A" after the prompt and then four hex digits of the memory location that you want to change. The Monitor answers you back by typing out the data that is presently in that location followed by a dash.

If you want to change the location you enter in the new data, or if it is OK and you want to examine the next location, all you need do is hit the space bar. If you hit a carriage return, you will get a carriage return, line feed, which will print out the address of the next memory location that you want to change.

To get back into Monitor, all you need do is hit the BREAK key on your keyboard. The Alter command is the normal way that programs are entered into memory from your keyboard.

C—Copy Memory to Memory. Another useful routine when it comes time to patch programs. With the use of this routine it is possible to move your programs around to any area in memory that you want.

The manner in which you use this command is putting a "C" after the prompt with the starting address and ending address of the program you are going to move. This is then followed by the beginning address of the area in memory where you are going to relocate that program to. There is no need to specify an ending address in this area since it is assumed that you will leave yourself enough area for your program to go into.

D—Dump. Useful when you want a listing of a specific program on your terminal. Here you use the starting address and then the ending address followed by a carriage return. You will get a hard-copy dump to your printer of the memory area specified. You'll see 16 memory locations to a line.

1. Alter memory
2. Breakpoint (set and reset)
3. Copy memory to memory
4. Dump
5. Go to users program
6. Hexadecimal arithmetic
7. Inspect CRU bit
8. Load
9. Modify CRU bit
10. Program EPROM
11. Snap
12. Workspace Dump

Fig. 7. Monitor commands.

G—Go to user program. The typical command of this type that transfers control of the processor from the Monitor to the program you want to run. Format for this command is "G," followed by the beginning address of the program you're going to run.

H—Hexadecimal Arithmetic. Allows you to calculate the sum or difference between two hex numbers. This is no more than hex addition and subtraction. Between your two numbers you either insert a plus or minus sign to indicate which

The Kit's documentation.

operation you are going to perform. This is, of course, preceded by the command "H." This command is extremely useful when you calculate jumps while you are programming.

I or M—Inspect or Modify a CRU bit. Used for looking at your Communications Register Unit, which has 16 input lines and 16 output lines. At any time you are allowed to examine or change these lines with the use of these commands. These are the main lines of input-output from the processor and are very useful in the control of external devices. The format for these commands is I or M, followed by the hex number for the line you want to change.

P—Program EPROM. Used for programming 2708s. Two sockets are provided—one for the even bytes and the other for odd bytes. When you program a PROM, it is extremely important that you specify a closing *even* address. The format for this command is "P," followed by the starting address then ending address you want to copy into the PROM and then a 0. In the Monitor for programming PROMs it automatically inserts the full address where the PROM programmer resides in memory (more on this later).

Finally, there are three commands that are used in conjunction with one another for debugging programs you have written: Workspace Dump (W), Breakpoint (B) and Snap (S). The Workspace Dump allows you to dump to your terminal a specific workspace area for examination. In this manner you can see if the proper data is being loaded into the correct registers.

The Breakpoint command can operate by itself or in conjunction with the Snap command. The most efficient way to utilize this Breakpoint command is with the Snap command as this allows you a readout of important information when you are debugging your own programs.

First you must set a Breakpoint somewhere in your program. To do this, type a "B," followed by the address and number of words to be removed for the trap. Next, initiate the Snap command by typing an "S" on your terminal.

The format for this command is to indicate the first time a snap is desired, then the number of increments between the snaps and finally the total number of snaps you want. The Monitor will answer you with an "R?" and in this area you will put the span of registers that you want printed out. Then the Monitor will give you an "M?" where you will enter the memory locations that you want printed out during each of the snaps. As you can imagine, these are extremely useful commands once you begin expanding your own programming expertise.

Observe one note of caution when you begin programming your own PROMs: As previously stated, the ending address of the area you want to program must be an even word. You must remember that you are programming in hex and that zero is considered an even bit. For example, if you wanted to program the information in location "0" to "F" into your PROMs location F100 (PROM address begins at F000), the format would be: "P 0000, 000E, 100 followed by a carriage return." Once those areas have been programmed, the Monitor will return with a "?" awaiting your next command.

Even though the Monitor is only 512 words in length, you can see that it is extremely useful and versatile. You receive all of the basics that you need for programming, debugging and running your own programs. The Monitor is easy to master . . . before you know it, it'll be second nature to you.

Instant Input Assembler

The Monitor, however, does have its idiosyncracies: Because of the aspect of microprogramming, all of your coding must be preconceived. This means you must sit down with pencil, paper and your program card to determine your machine's instructions for that program. You must code in your source and destination registers, your mode of addressing and distance of relative jump, all of which must be in hexadecimal machine language.

If you are already familiar with microprogramming machine-language instruction and enjoy it, then read no further. If you're like me, you'll be pulling your hair out and getting nowhere fast trying to microprogram. Alas, there is salvation, which comes in the form of a PROM known as the Instant Input Assembler.

This miniature assembler is a stroke of genius and must be purchased as an extra option. But the one fact that cannot be denied is that it makes programming so much easier as compared to machine level language that you won't believe it. You are able to use the op codes and register designations, and the assembler then decodes this information into machine language and outputs to your printer.

It is a joy to use and simplifies programming the Super Starter Kit beyond belief. It is compact (512 words in length) and packed with features that make working with this kit a dream. You are allowed a String Constant, Numeric Constant, Address Redefinition, Use of Instruction Mnemonics, Register Definition and can define all Addressing Modes.

To begin operation of the assembler you must first be in Monitor. After you receive the "?" from Monitor, you enter "G F800," followed by a carriage return from the keyboard. This is the starting address of the assembler. The assembler will answer with a carriage return, two line feeds, 0100: and seven spaces. The "0100" is the first memory location to which data is going to be entered and the spaces are going to be for the hexadecimal value of the information entered on this line.

Upon completion of data entered, you hit the carriage return key on the keyboard. You will see the assembler print out the hexadecimal value of your information followed by a carriage return, line feed and the address of the next memory location in which data can be entered. It is much easier to enter your programs using the assembler, as you will see shortly.

The manner in which you enter your data can sometimes vary. You should always have your manual nearby to be sure whether or not an instruction can perform a desired function. In the instruction set section of the manual there are some examples that show how instructions can be employed (see Fig. 8).

There are numerous formats you can use with this assembler as well as "Line Editing and Error Message Printouts" when you make a mistake. Fig. 8 is taken verbatim from the "Assembler Section" of the manual to show the clarity of the documentation. It is reprinted here with the permis-

Format: MOV S,D
Opcode: C000
Status Changed: LGT,AGT,EQ
Definition: Replace destination operand with a copy of the source operand.
Results: (S)→(D)
Notes: Use to move from:

Memory to Memory	MOV @TABLE,@TEMP
Register to Register	MOV R5,R9
Register to Memory (Store)	MOV R3,@ANSWER
Memory to Register (Load)	MOV @TABLE,R8

B. Entry Format

When the assembler is awaiting input, you may enter any of the following types of commands:

1) string constant—Type a "$" followed by the characters that you want entered in memory. The assembler will convert each character from ASCII to hexadecimal and store the constants in memory. The string may be any length and is terminated by a carriage return. If the string is an odd number of bytes long, the assembler will add one more byte filled with a space to return to an even address.

2) numeric constant—A numeric constant is indicated by typing a "+" or "−" followed by the desired constant. All constants are decimal unless otherwise indicated. To indicate a hexadecimal constant, precede the number by a ">" (e.g., + > 1234). To indicate a binary constant, precede the number by a "%" (e.g., + %10100011100)

3) address redefinition—You can change the program counter location by typing a "/" followed by the new address (in hexadecimal). This feature is ideal for patching because it allows you to move about in memory without restarting the assembler.

4) TI 9900 assembly instruction—All of the TI 9900 mnemonics are recognized by the instant input assembler. The general format of an instruction entry is:

<instruction-mnemonic> space <operand-field> space

The complete set of allowable instruction mnemonics is shown in the Super Starter Kit. There are several different operand fields that may be used:

a) register—for example R0, R13, 12, 3
b) register indirect—for example *R15, *6
c) register indirect with auto increment—for example *R14+
d) indexed—for example @12(R1), @123, @%101, @>12(3)
e) constant—for example 12, %101, >123A
f) string—for example 'A', 'BD'
g) relative displacement (for jumps)—The displacement is in words and is checked for allowable range. There are three possible formats: +N, −N, or N. The +/− format is a jump forward or backward N words from the *next* sequential word. The N format is a jump to address N, the assembler will calculate the offset in this case.

Refer to the instruction manual to determine what operands are allowed with any given instruction. Sample instructions and mnemonics are:

C. Line Editing

If you make a typing mistake, you can back space and correct it by typing a back space (CTRL-H). The assembler will line feed and back up (assuming your terminal can back space) under the character to be changed. If your mistake is too big, type ESC (escape) and the assembler will discard this line and allow you to start over again.

D. Error Messages

The instant input assembler is only 512 words long, so it cannot detect all user errors; however, it does detect most of them. Whenever you type something that the assembler does not like or understand, it will type an error message on the next line and reprompt with the old address. The possible error messages are:

*S—syntax error. The input contains a syntax error.
*D—displacement error. The target address of a jump is too far away and exceeds the allowable range.
*R—range error. The input is out of range. It should have been 0 to 15.

When the assembler detects an error and prints the message, it keeps the program counter set to the location of the error. That is, the program counter is not advanced until a statement is accepted and stored in memory.

Fig. 8.

?A0100	FF-02	C3-01	C4-01	D3-25	01-2C	45-71	C3-C0	28-91	45-04	36-60	37-01	65-04
LOCATIONS	100	101	102	103	104	105	106	107	108	109	10A	10B

Example 1.

sion of Technico.

As you can see, it is well detailed and explained in a clear and concise manner. An additional feature that is incorporated into this assembler is the manner in which the I/O is handled. All that you need to do is specify "IN" to accept data from your keyboard or "OUT" to transfer data to your terminal.

These two additional instructions are not located in the instruction set as they are special options built into the assembler. Remember, though, you must still follow the assembler format even with these instructions.

Detailed below are two examples that act as a comparison between the use of just the Monitor and the assembler when a program is entered.

As previously stated, when using the Monitor you must first sit down and figure out exactly what the hexadecimal value of your instruction is going to be. The short program shown in Fig. 8 accepts data from your keyboard and stores it in successive memory locations.

To do this, first set up a register to point at the memory location where you want to put the data. This is done by loading a value into register #1. Use the "Load Immediate," Op Code 0200 and instruction format #8. Fig. 2 shows that this would work out to be 0201 hex followed by the location where you would put the data.

Now that you have set up the starting location of where you are going to store your data, you must somehow get the information there. Since the input to your machine utilizes the "XOP" Instructions #1, you have to code this into your machine. But you don't want to store directly into the register; you want to go indirectly and then increment register 1 by one. Again looking at the Instructions, you would see it uses Format #9, and the code would be "2C71."

The next step would be to Move the Register to be ready for new data to pass through. The Move Instruction uses Format #1 and the code would be "C091." Finally you have to Branch back to your input area of the program, which is the XOP Instruction. The Branch Instruction uses Format #6 and places the specified value into the program counter. The code for this would be "0460" followed by the location where your XOP Instruction would be located.

After you have done all these pre-calculations you would then go to the machine, enter the Monitor and enter your code with the Alter command. This would be done as shown in Example 1 if the starting address of your program were "0100." (Note: The first hex value of each location is random information that was in that location. The information you are programming into the 9900 is after the dash.)

As you should be able to see, if you had a long program, this manner of programming could take considerable time with all of the pre-planning that must be done. With the Instant Input Assembler, your pre-calculations are kept to a minimum and programming is much simpler and more enjoyable. With it, all that you need do is to figure out the instructions you want to use, keep track of registers used and addressing modes and then enter your mnemonics.

I knew I had to set up a register for a specific value so I picked register 1 with the Load Immediate Instruction. Input was Indirect with an Auto Increment. I Moved R2 indirectly to R1 and Branched back to my Input Instruction. After I had this basic outline of my program, I went to the 9900 and entered the assembler area. See Example 2. (Note: The program counter is always the first set of numbers specified on the line. The underlined information is what the assembler prints out following the carriage return after the Mnemonic Instructions.)

The value and power of the Instant Input Assembler should now be apparent. Programming is made much easier with it. But don't forget about the debugging capabilities of the Monitor for ironing out any programming mistakes. Together, both of them give you a powerful programming tool.

Conclusion

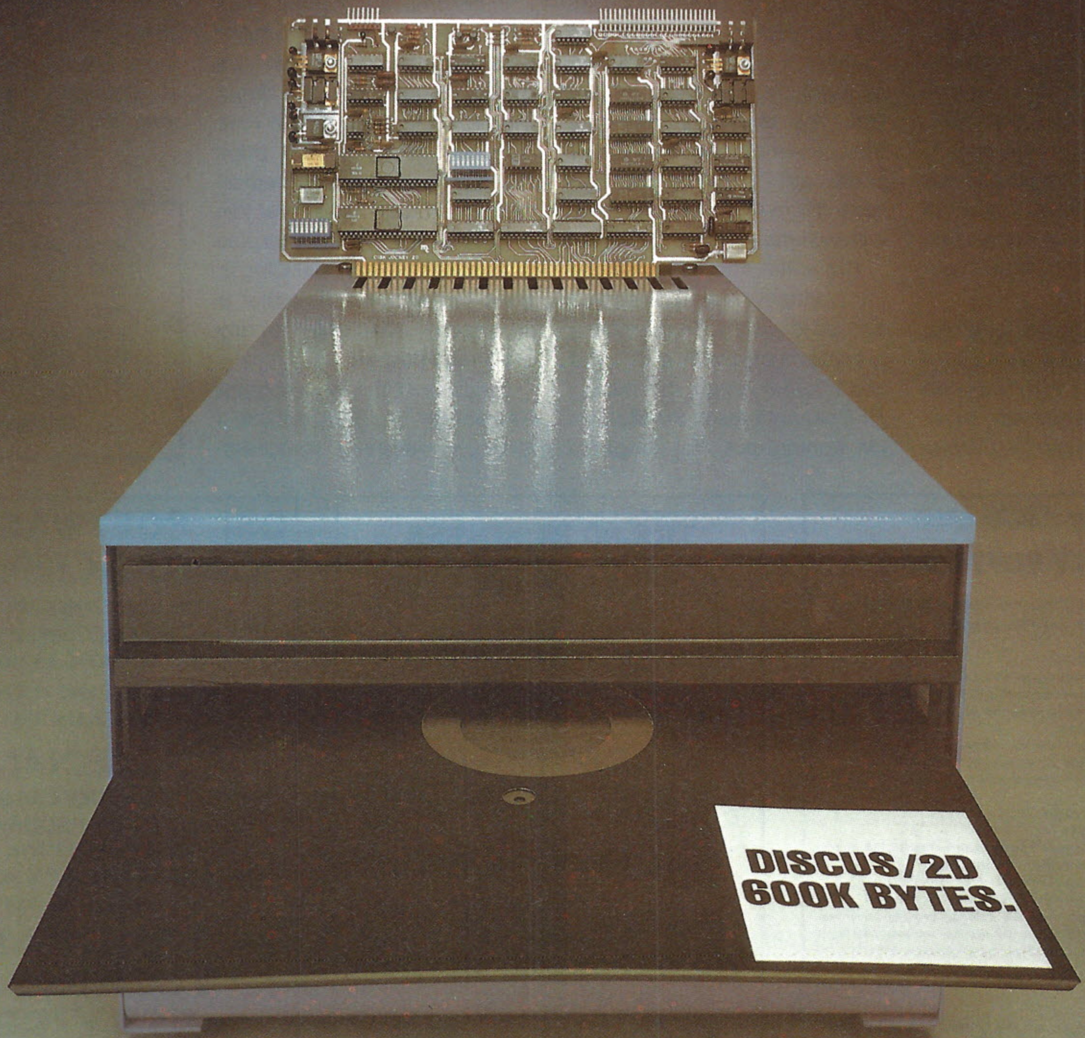
In closing, let me state that I have tried many of the major 8-bit microprocessors on the market today. But the Super Starter Kit is the one that I am truly impressed with. I think that it is the best one of them all and one of the easiest that I've worked with. I had little trouble in assembling it or operating it. It is a powerful machine that is universally adaptable to any peripheral device.

Because of the Instant Input Assembler (a $49 option I strongly suggest), programming is fairly easy to learn. Though this is the first 16-bit hobbyist mini/micro machine that I have worked with I have found it easier than many of the 8-bit machines. I cannot say enough about Technico's Super Starter Kit, except that it is a dream machine come true. ■

```
0100:  0201   LI R1,>125
0102:  0125
0104:  2C71   IN *R1+
0106:  C091   MOV *R1,R2
0108:  0460   B @>104
010A:  0104
010C:
```

Example 2.

DOUBLE DENSITY



SOLID SAVINGS!

Now you can put your S-100 system solidly into a full-size, single/double density, 600K bytes/side disk memory for just $1149 complete.

DISCUS/2D™ single/double density disk memory from Thinker Toys™ is fully equipped, fully assembled, and fully guaranteed to perform perfectly.

DISCUS/2D™ is a second generation disk memory system that's compatible with the new IBM System 34 format. The disk drive is a full-size Shugart 800R, the standard of reliability and performance in disk drives. It's delivered in a handsome cabinet with built-in power supply.

The S-100 controller utilizes the amazing Western Digital 1791 dual-density controller chip . . . plus power-on jump circuitry, 1K of RAM, 1K of ROM with built-in monitor, and a hardware UART to make I/O interfacing a snap.

The DISCUS/2D™ system is fully integrated with innovations by designer/inventor George Morrow. Software includes BASIC-V™ virtual disk BASIC, DOS, and DISK-ATE™ assembler/editor. Patches for CP/M* are also included. CP/M*, MicroSoft Disk BASIC and FORTRAN are also available at extra cost.

DISCUS/2D™ is the really solid single/double density disk system you've been waiting for. We can deliver it now for just $1149. And for just $795 apiece, you can add up to 3 additional Shugart drives to your system. Both the hardware and software are ready when you are.

Ask your local computer store to order the DISCUS/2D™ for you. Or, if unavailable locally, write Thinker Toys,™ 5221 Central Ave., Richmond, CA 94804. Or call (415) 524-2101 weekdays, 10-5 Pacific Time. (FOB Berkeley. Cal. res. add tax.)

*CP/M is a trademark of Digital Research.

Morrow makes disk memory for

Thinker Toys™

Thoughts on the SWTP Computer System

The third installment in this series looks at the MP-S and the MP-C interfaces.

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

This is the third in a series of articles on the SWTP 6800 computer. One of the purposes of this series is to spark some interest in this fine system. From some of the reader mail, it appears that most of us are pleased with our SWTP computers.

The MP-S vs the MP-C

What are the differences between the MP-C control interface and the MP-S interface? And if you have an older system with an MP-C, should you update to the MP-S? Let's look these interfaces over.

Similarities. Both interfaces plug into one of the I/O slots in the system and provide an interface for a serial terminal such as a teleprinter or a CRT terminal. With appropriate software changes, they can be plugged into any of the eight ports, but most of us have them in port 1, since this is defined as the control port. This is the port that the Monitor, BASIC and all other system software use as the main input and output port.

In addition to providing input and output interfacing for either a current loop or an EIA RS-232C system, both also provide an additional "reader on" control signal that can go to a relay in a Model 33 Teletype machine to control the paper tape reader. Nowadays, most of us use that signal to control switching between the keyboard and the cassette input.

Differences. The MP-S achieves the conversion from the parallel data form in the computer to the serial form required by the terminal by using an MC6850 ACIA (asynchronous communications interface adapter). This is a specialized form of a UART that is specially designed to interface all of the standard UART functions directly to the MC6800 data bus. Whereas a UART requires several external control signals and usually also needs external strapping of some pins to either ground or +5 volts to set the number of stop bits, the number of data bits and parity type, the ACIA allows these functions to be programmed by the computer via the data bus.

In the ACIA, the conversion to and from serial is accomplished by hardware, usually at the same time the processor is doing something else. To output a character to the terminal, for instance, the 6800 simply sends a character out to the ACIA and then proceeds to prepare the next character while the ACIA takes care of sending the present one out. Thus I/O can be overlapped with computing.

The MP-C control interface, on the other hand, uses an MC6820 or MC6821 PIA (peripheral interface adapter). This is a parallel device, capable of providing two full 8-bit ports with complete handshaking for each. (This is the same IC used in the MP-LA parallel port interface, which is often used to interface the PR-40 parallel printer.)

Using a PIA to provide serial I/O is an awkward way of doing the job, but it dates back to SWTP's use of Motorola's MIKBUG monitor. (MIKBUG and SWTBUG are trademarks of Motorola, Inc., and Southwest Technical Products Corp., respectively.)

MIKBUG was developed at a time when the ACIA had not yet been developed, and so Motorola used the PIA for serial I/O. Since SWTP used the MIKBUG monitor until they developed their own SWTBUG, they had to carry out the I/O the same way Motorola did. Hence they designed a special control interface, the MP-C, just for this purpose.

MP-C Disadvantages

Since the 6820 PIA does not perform its own conversion from parallel to serial and vice versa, the processor must do it with software. The program for this conversion is part of MIKBUG and SWTBUG. Although this is completely transparent to the casual user, it does affect all of us in that it slows down the operation of most programs. That is, when a program is output through the MP-C to a terminal, the computer is completely tied up with the timing, and hence cannot be working on anything else.

To see what this means, let's look at an example. Suppose we are printing at 300 baud and the program needs 30 milliseconds (ms) to ready each character for printing. At 300 baud, each bit takes 1/300 second, which is 3.33 ms, to be sent; an entire character takes 33.3 ms.

With an MP-S serial interface, the processor simply sends a character out to the ACIA and starts it; it then has 33.3 ms to get the next character ready before the ACIA wants it. Since it only takes 30 ms in our example, the processor will be ready with it by the time the ACIA wants it and will be able to send it out right away. Hence data will be going out at the full 33.3 ms per character rate, just as if the processor needed no time at all between characters. The processing time simply overlaps the I/O time.

But with an MP-C interface, while one character is being

sent out the processor is busy timing it . . . hence it cannot be working on getting the next one ready. When the 33.3 ms for one character have elapsed, the processor will then spend 30 ms getting the next one ready before being able to send it. Thus the characters will be coming out only at the rate of one every 63.3 ms, instead of one every 33.3 ms. This works out to only 16 characters per second instead of 30. Thus, in this case, the printout would take almost twice as long with an MP-C as with an MP-S.

This doesn't really happen very often, because usually the processor doesn't need that much time between characters. Generally it will need only one millisecond or less, so with an MP-C the characters would come perhaps 34 ms apart, while with an MP-S they might come 33.3 ms apart. The difference is hardly noticeable most of the time . . . and the slower the baud rate, the less the difference. At 110 baud you might never notice it at all.

But this effect does show up in three specific ways. First, it determines the maximum data rate, which is about 1200 baud for the MP-C. At higher baud rates, the processor simply cannot adequately time each bit and control it accurately. The MP-S can be used much faster than that, since the ACIA does its own timing via an external clock.

Second, let's look at stop pulses. A normal ASCII serial character consists of the following (see Fig. 1):

1 start bit, always a 0
8 data bits, either 0 or 1
1 or 2 stop bits, always a 1

At 110 baud, which is the speed often used by mechanical teletypewriters, two stop bits are required to give the printer mechanism time to de-energize a clutch between characters. At speeds of 300 baud and up, one stop bit is usually enough since at these speeds the characters are decoded by electronic circuits rather than by clutch-driven distributors, and so the extra stop bit is not needed. Although just one stop bit is sufficient at these speeds, it doesn't hurt to include a second stop bit. It simply means that whatever is receiving the serial data has to wait a little longer before getting the following start bit.

This kind of serial data is called *asynchronous*, which means that the characters are not arriving at regular intervals. It's understood that a character could arrive just after the previous one ends, or any time after that. There might be a microsecond or an hour between them. The time difference is unimportant, and the signal sent between characters is a 1, the same as a stop pulse. We can think of this as just one long stop pulse. In other words, as long as it's at least one bit-time long (3.33 ms at 300 baud, for instance), the length of a stop theoretically doesn't matter.

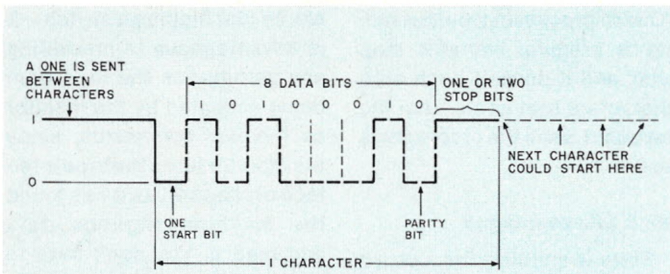


Fig. 1. A typical ASCII character in serial.

When you use an MP-S interface, in most cases the processing time between characters is short compared with the I/O time, and so each character will immediately follow the one before it. Hence the stop bit will be exactly one bit long (or two, if the ACIA has been programmed for two). The characters will arrive like clockwork.

But when an MP-C is used, the processing time will be added to the character time, and there will always be extra time needed between characters for the processor to get the next character. This might be just a fraction of a millisecond, or it might be a relatively long time. This time is added to the stop bit-time and simply extends it to more than its usual length. Since it is unlikely that the stop bit will be lengthened by exactly one bit-time, it will usually be some weird multiple of its normal length. In other words, the timing between characters now becomes uneven.

This can happen with an ACIA too, except that seldom will the processing time for each character be long enough so that it can't be accomplished while the ACIA is busy. So it's not likely to happen with the ACIA as with the PIA. In most cases, this doesn't matter. As long as the stop bit is longer than its minimum length, most I/O equipment will operate smoothly.

There is some equipment that requires the appropriate stop bit-time: an exact multiple of a single bit-time (6.66 ms or 9.99 ms, for instance, at 300 baud). But when the stop bit is some weird multiple of a bit-time, problems occur.

More on this next time, but for now let's just say that here is one case where an MP-C interface can create some sizable problems. It's a problem that can be eliminated by careful programming, since the processor could simply stretch out the stop bit to make it an exact multiple of the correct time, but neither MIKBUG nor SWTBUG is that careful.

There is a third source of possible problems with the MP-C. Just as the MP-C needs extra time between characters during output to prepare the next character, it also needs extra time between characters during input to put the received character away into memory wherever it has to go.

Suppose you have a cassette tape that was recorded with an MP-S ACIA interface, so that each character on the tape follows immediately after the one before. When read with another ACIA, the processor will have time to take care of each character as the next one is being read. But when a PIA is used for input, the processor may miss the beginning of the next character if it takes too long with the previous one.

Changing the Stop Bits

There is a simple solution to this problem—record all tapes with two stop bits but set the interface for only one stop bit on input. Hence there will always be an extra stop bit's delay between characters, during which time the processor can work on the previous character.

There is a second good reason for this. Since the Kansas City tape standard recovers its clock bits from the tape data, a dropout on the tape may lose some clock bits. If the data on the tape is recorded with only one stop bit, this may cause the entire tape to go out of step with the clock. All the subsequent data will be read wrong. But if each character on the tape has an extra stop bit, the extra clock pulses in the stop bit will let the computer (and terminal) catch up within a few characters. The rest of the tape will then read correctly.

On the MP-C, a jumper on the board sets the number of stop bits. This jumper goes from point C on the board to either the 110 hole (for two stop bits) or the 300 hole (for one stop). In my system, I bring this jumper out to a switch on the front panel. Normally I keep it set to one stop bit but flip it to two whenever I record a tape.

One company that has done some work in this area is Percom Data Company, Inc., 318 Barnes, Garland TX 75042, since it is related to their CIS-30+ cassette interface. If you are interested in their story, order the following technical memoranda from them:

TM-CIS-30-02 ($.50). Using the Percom CIS-30+ with SWTP 8K BASIC Version 2.0.

TM-CIS-30-03 ($.25). Problems with the MIKBUG (TM) Input/Output Routines.

TM-CIS-30-15 ($.25). Software Hints for Using the Percom CIS-30+ with ACIA I/O Interfaces. (This is a general discussion of

ACIA programming and does not apply to SWTBUG without some changes.)

It might be a good idea to enclose a large SASE with your order.

The solution they recommend for the MP-C is to connect the stop-bit jumper from point C to IC5 pin 3. This is the reader control pin, which is normally high but goes low when the monitor tries to load data. This means that the MP-C is always set to two stop bits but will now automatically switch itself to one stop bit whenever it is loading under monitor control. Since most system programs such as BASIC don't turn the reader on, this will not work for them, but Percom suggests several patches that will solve the problem.

We are better off with the MP-S. Changing the number of stop bits the ACIA generates during output (or expects during input) has to be programmed into it by the processor. Fortunately, SWTBUG does this for us. Whenever SWTBUG executes the INEEE input subroutine, it immediately switches the ACIA to expect one stop bit. Every call to OUTEEE, on the other hand, switches the ACIA to two stop bits. As long as we use INEEE and OUTEEE in every program, SWTBUG neatly solves the problem (and introduces another one, as we will see).

More MP-S Advantages

The MP-S has one further advantage over the MP-C. When a character comes in, the ACIA in the MP-S will catch it and hold it for the processor, even if the processor is busy at the time. In the same situation, the MP-C would simply miss it.

This makes a big difference in how fast BASIC responds to a control-C. With the MP-S, hitting a control-C while BASIC is running stores the character in the ACIA; when BASIC gets around to looking for it, it is still there. Hence hitting control-C just once is enough to stop BASIC. But with the MP-C, unless you happen to hit the control-C at the exact instant that BASIC is looking for it, it will be missed; you may have to hit the control-C hundreds of times before BASIC finally stops.

Let's review MP-C disadvantages: It's slower (limited to 1200 baud); it doesn't overlap I/O with processing; output timing is irregular (variable stop bits); and it doesn't latch onto characters that come from the keyboard while the processor is busy.

MP-S Disadvantages

From everything I've said so far, it would seem that I think the MP-S is a tremendous improvement over the MP-C. This is not quite true. To my mind, the MP-S has some disadvantages too.

So let's look at the MP-C's good points. First, MIKBUG does not support the MP-S. Unless you switch to SWTBUG, you must stick with the MP-C if you still use MIKBUG.

Second, both the MP-C and the MP-S provide a reader control line for turning on the teleprinter's paper tape reader or switching the cassette interface from keyboard to tape. But only the MP-C actually does it under MIKBUG or SWTBUG control. This is really not an MP-S problem; it's a software problem. But short of burning your own monitor into an EPROM, using a monitor in RAM, waiting until SWTP—we hope—comes out with an SWTBUG II or using someone else's monitor, there isn't much you can do about this one.

With the MP-C and either MIKBUG or SWTBUG, the L command turns on the reader control line at the start of reading and turns it off when done. In this way, the cassette interface automatically switches itself on and off.

Although this seems like a trivial matter—after all, we could do the same thing manually by just flipping a switch—it is advantageous in preventing any garbage on the tape from being accepted by the monitor as monitor commands, since the monitor turns the tape interface off as soon as it has found the machine-language data and read it. You don't have to keep staring at the terminal to catch the first piece of garbage that comes and quickly turn off the tape.

With the MP-S and SWTBUG, we find that SWTBUG doesn't turn the reader on or off at all. We have to do it manually. Not only that, we can't even conveniently write a program to do it, because SWTBUG will cancel the command!

As I mentioned earlier, SWTBUG automatically switches to one stop bit with INEEE and to two stop bits with OUTEEE. The same instructions that regulate this also keep the reader control line turned off. So even if we turn the reader on, as soon as we use INEEE it gets turned off again.

If you have SWTBUG and want to see where it's performed, look at locations E1FB and E212. In E1FB the sequence is

LDA A #$15
STA A 0,X

which stores a $15 into location 8004, the ACIA control register. Likewise, the E212 sequence is

LDA A#$11
STA A 0,X

which stores a $11 into 8004. The second digit determines the number of stop bits; a 1 means two stop bits, and a 5 means one.

Unfortunately, the first digit (the 1) turns off the reader control; to turn on the reader, E1FB would have to be changed to LDA A #$55. There is no way to modify SWTBUG. But if you have your own monitor either in RAM or in a 2716 EPROM on your CPU board, you can put in a different INEEE routine. See Program 1 for the INB routine, which is part of a custom monitor I use. It is used instead of INEEE in a binary loader, which is also in the same EPROM.

Another advantage of the MP-C PIA-type interface is that it is primarily a parallel interface, which has two parallel ports and four handshaking lines. Only some of these are used for the serial interface, so that several lines are still available for other use.

Each time we reset, the monitor initializes the two ports as follows:

Port A—Bit 0 (least significant): output
Bits 1 through 7: input
Port B—Bits 0 through 2: output
Bits 3 through 7: input

Of these pins, bits 0 and 7 in port A and bits 0, 2, 6 and 7 in port B are used for the serial interface. This leaves ten unused bits, of which one is output and nine are configured for input. (The SWTBUG manual claims that four of the bits on port A are used for controlling a tape reader and punch, but they are configured as input pins and cannot do that job without more programming.)

Thus we have six pins on port A that could be used for inputs to sense external events, plus three more input pins and one output pin on port B. (It's possible to use the port A pins for inputs or to reconfigure them for output, but be careful since SWTBUG may interfere with that.)

Another possible use for the extra MP-C bits is for serial Baudot data for a five-level teleprinter or even for Morse code. (The ACIA in the MP-S can't be

```
CA3B 86 55    LOADB  LDA A  #$55     TURN ON TAPE AND 1 STOP BIT
CA3D B7 8004         STA A  ACIA1
```

A. Initializing the ACIA to turn on Reader Control.

```
               *INPUT DRIVER
CA89 B6 8004 INB      LDA A  ACIA1     GET ACIA STATUS
CA8C 47               ASR A            SHIFT READY BIT
CA8D 24 FA            BCC    INB       REPEAT IF NOT READY
CA8F B6 8005          LDA A  ACIA1+1   GET CHAR WHEN READY
CA92 39               RTS
```

B. ACIA input driver to replace INEEE and leave Reader Control on.

Program 1. ACIA program to use Reader Control line.

used for five-bit codes such as Baudot.) The timing would have to be controlled by software, but the use of these bits would enable us to add more functions without adding more interfaces. (The MP-C seems especially well-suited for Baudot data, since it already has all the circuitry for current loops, and only a software change is needed to use it. But be careful—most Baudot teleprinters do not have the magnet driver card that a Model 33 or 35 ASCII printer has and, therefore, will not operate satisfactorily from the low-voltage MP-C output without an external loop driver. The only Baudot teleprinter that could be connected directly would be a Teletype Model 32.)

Some users have also used the extra PIA inputs to detect the state of sense switches mounted on the front panel. This can be useful for alternate routines, depending on the state of the sense switch.

One other advantage of the MP-C is that by simply changing the software, we can change the data baud rate without changing the baud rate clock input to the MP-C. One cassette interface manufacturer takes advantage of this to record data at 2400 baud. (See the cassette interface reviews in the next installment of this series.)

Hardware vs Software Echo

One more difference between the MP-S and MP-C is how they handle terminal echo. In the MP-C, there is a connection from the keyboard input back to the CRT or printer output to provide for echoing entered characters back to the terminal. This echo signal goes through a gate that is controlled by the reader control signal. Whenever the processor turns on the reader, it also disables the echo. The only way to turn off the echo is to turn on the reader, which may not always be desirable.

This can cause you trouble if you simultaneously write a tape and accidentally hit a key on the keyboard. The keyboard character will interfere with the tape data, and the tape will contain an error.

In the MP-S, on the other hand, there is no hardware echo. Instead, echoing is accomplished by the processor receiving a character from the input and then sending it back out the output. When the MP-S is outputting to tape, the echo is off, so no garbage is recorded on tape if you hit a keyboard character. Echoing and reader control are completely independent.

But it has some other side effects. First, most terminals have a HALF-FULL switch that provides for an internal echo. This switch also sends keyboard signals back to the CRT or printer when in the HALF position. With an MP-C it doesn't matter whether this switch is in HALF or FULL, since either way the keyboard signal goes right back to the display or printer.

But with an MP-S, the echo character from the MP-S doesn't get sent out until the keyboard character has been received. Thus there is a delay. If the switch is in HALF, then the display or printer will receive an echo immediately through the terminal's HALF-FULL switch, and another echo right afterward. Thus each character will be printed twice.

With the MP-C, the echo is exactly the same as the character that went in, complete with timing. If the terminal and the computer are at different baud rates, the echo will be at the terminal's baud rate rather than at the computer's. Hence the terminal will print the correct character even though the computer didn't understand a bit of it. Somewhat misleading.

With both interfaces, if the input comes from a cassette recorder whose speed is too high or too low, the data will arrive at the computer too fast or too slow. Since the computer's baud rate clock is derived from the tape, it will be received correctly. But if the terminal operates from its own clock, it may print garbage even though the computer understood it all. Again somewhat misleading.

The MP-S has another interesting feature. Both the input clock and the output clock come from the same source (pins 3 and 4 of the ACIA are jumpered together), and hence the data being echoed to the terminal will run at the same incorrect speed as the tape. But if the connection between pins 3 and 4 is broken and the output clock is always taken from the computer clock, then the computer can be used to reclock the signal, and the terminal will copy the data correctly if the computer does. (Though if the tape speed is too fast, the characters may be coming in faster than the computer can echo them, and then occasionally a character may be missed. This is less likely to happen if the tape is recorded with two stop bits, since this slows down the incoming characters by about 10 percent.)

Using Both an MP-S and an MP-C Interface

The upshot of all this is that, even if you decide to switch to the MP-S interface to take advantage of its higher data rate, it may pay to keep the MP-C in port 0 if you have a separate cassette interface or printer to connect there.

This operation is supported by the O command in SWTBUG. When the L, P or E command is preceded by an O, SWTBUG performs the Load, Punch or End command, but with an interface in port 0 rather than port 1. (The SWTBUG manual says that this port 0 interface must be an MP-C, but that is not true—it will work for an MP-S too.) Interestingly enough, SWTP 8K BASIC Version 2.3 also supports an MP-C in port 0.

In my system, I have two different cassette interfaces—one on each port. I also have a CRT terminal on port 1 that I use at either 300 or 1200 baud, and a 300-baud serial printer on port 0. To allow the use of port 0 as an input port, the keyboard output of the CRT terminal is fed in parallel to both port 1 and port 0. Its data will then be accepted on either port. (I have to switch the terminal's baud rate to 300 if I want it to input to port 0.)

This is actually a good arrangement, and I recommend it to anyone who has both a CRT and teleprinter, since it solves the speed-mismatch problem—the CRT can be operated fast, and the printer slow.

To make all this work out requires some patches to software. I have a 2716 EPROM on my CPU card that has a number of subroutines which drive port 0 (see Program 2). These are IZPTO, which initializes the PIA, OUTPTO, which is used exactly like OUTEEE, and INPTO, which replaces INEEE. IZPTO is not really needed, since the PIA is initialized by SWTBUG whenever you type 0E into the monitor, but the existence of a separate routine avoids problems in case I forget to initialize it

```
                *
                *INITIALIZE PORT 0 SUBROUTINE
                *
CB04 7F A00B IZPTO  CLR    PORADD+1 CHANGE 8004 TO 8000
CB07 FE A00A        LDX    PORADD
CB0A BD E284        JSR    PIAINI   INITIALIZE PIA
CB0D BD E27D        JSR    PIAECH   SET UP ECHO
CB10 86 04          LDA A  #4
CB12 B7 A00B        STA A  PORADD+1 CHANGE 8000 BACK TO 8004
CB15 39             RTS
                *OUTPUT TO PORT 0 PIA
CB16 7F A00B OUTPTO CLR    PORADD+1 CHANGE 8004 TO 8000
CB19 BD E1D1        JSR    OUTEEE   PRINT ON PORT 0
CB1C 86 04          LDA A  #4
CB1E B7 A00B        STA A  PORADD+1 CHANGE 8000 BACK TO 8004
CB21 39             RTS
                *INPUT FROM PORT 0 PIA
CB22 7F A00B INPTO  CLR    PORADD+1 CHANGE 8004 TO 8000
CB25 BD E1AC        JSR    INEEE    INPUT FROM PORT 0
CB28 36             PSH A           SAVE THE CHARACTER
CB29 86 04          LDA A  #4
CB2B B7 A00B        STA A  PORADD+1 CHANGE 8000 BACK TO 8004
CB2E 32             PUL A
CB2F 39             RTS
                *GO TO PROGRAM WITH PORT = 0
CB30 8D D2   GOPTO  BSR    IZPTO    INITIALIZE PORT 0
CB32 7F A00B        CLR    PORADD+1 CHANGE 8004 TO 8000
CB35 FE A048        LDX    $A048    GET STARTING ADDRESS
CB38 6E 00          JMP    0,X
```

Program 2. Routines for using an MP-C on port 0.

manually.

In addition, the EPROM has a routine called GOPT0, which is similar to the monitor's G command (except that it does not handle the stack quite the same). This one sets up the monitor's port address for port 0 and then jumps into the program whose starting address is in locations A048 and A049. This technique does not work for some of the SWTP system programs such as BASIC or CORES, which do their own I/O independent of the monitor, but for most other programs it simply switches all output and input to port 0. The OUTPT0 and INPT0 routines need not be used in this case.

SWTP 8K BASIC version 2.0 is easily patched to use these port 0 routines by changing the I/O vectors as follows:

```
0106 JMP OUTPT0
0109 JMP INPT0
010C JMP IZPT0
```

Once this is completed, port 0 can be called with LIST #0, PRINT #0 or INPUT #0 instructions.

The easiest way to patch the CORES Editor/Assembler is by placing a jump to OUTPT0 in location 1A83. This retains port 1 as I/O but replaces the port 0 printer for the PR-40 printer on port 7. Port 0 then works when you type in the PRINT command.

Having a second serial interface like this is useless unless you have something to connect to it. But if you have a serial printer separate from your terminal, this does simplify many programs. For instance, BASIC programs can now operate through the CRT terminal and display all output there, while printing only selected portions. This can then be accomplished under program control.

If you have two cassette interfaces, one on each port, you can use one for input and the other for output and update cassette files. You can also now copy tapes from one interface to the other and have them retimed by the computer in the process.

Printer Interrupts

A good way to end this month's installment is to stay on the subject of I/O. Specifically, let's talk about using overlapped I/O with the PR-40 printer.

The PR-40 has a 40-character buffer, which can store an entire line before printing it. A program can fill up an entire line's worth of characters in the buffer at high speed and then go on computing while the printer outputs the line.

But when many short lines have to be printed, the computer may spend quite a bit of time waiting for the printer to output each line.

When this happens, you can speed things up quite a bit if you set up a larger buffer in memory, filled by the program at high speed and emptied by the printer as needed. You can do this with printer interrupts.

The idea is to set up a buffer in memory as a FIFO (First In, First Out) memory. The program can fill the FIFO as fast as it needs (unless it becomes full), and the printer can empty it at its own speed. This evens out the operation of the printer and keeps it operating constantly, rather than sporadically starting and stopping.

I mention this here because Applied Microcomputer Systems (Box 68, Silver Lake NH 03875) has a fairly inexpensive program (under $10 with Kansas City cassette) that does just this. It is a patch for SWTP 8K BASIC version 2.0. Although I haven't tried it, they claim that it speeds up programs as much as a factor of two.

Although interrupt lines are on all the SWTP I/O cards, standard SWTP software doesn't use them. This sure helps keep things simple, but it does mean that we don't use some of the advanced techniques that large computers have used for years. This interrupt printer driver seems interesting to experiment with. I will include an interrupt routine for a serial terminal on port 1 in a future article.

Conclusion

Next month we will examine four different cassette interfaces you can buy for your SWTP system.■

✓ Reader Service—see page 179

It's already a big success!

The party's over for all dumb terminals and a lot of smart ones too. But, at $798 (quantity 25), the party's just beginning for Intertec's InterTube II.

Standard features to celebrate include a full 24 line by 80 character display, 128 upper and lower case ASCII characters, reverse video, complete cursor addressing and control, an 18 key numeric keypad, special function keys, blinking, protected fields, character and line insert/delete, editing, eleven special graphics symbols, a 25th status line which displays the terminal operating mode and an RS-232 printer port.

You'll discover even more reasons to celebrate when you sit down in front of an InterTube II. Our wide bandwidth monitor produces crisp, sharp chararters everywhere on the screen. InterTube's Z-80 processor enables a host of operator oriented features to boost the efficiency of both software and programmers. And, InterTube's rugged modular design combined with its built-in self-test mode insures quick and reliable servicing.

InterTube's price/performance ratio can satisfy your requirements whether they be a sophisticated data entry application or a simple inquiry/response environment. So, there's really no reason to think "dumb" when you can afford to be so smart!

Join the thousands of InterTube celebrations going on around the country at this very moment. Call us at the number below and start your own celebration (BYOB—we'll bring the InterTube).

INTERTEC DATA SYSTEMS® 121

2300 Broad River Road, Columbia, S. C. 29210
(803) 798-9100 TWX: 810-666-2115

* $995 Quantity One

Instant Software™ Has It All...

Ask for Instant Software at your local computer store, use the handy order blank provided at right, or order your software by phone—call Toll Free 1-800-258-5473.

for the TRS-80**

Level I

STATUS OF HOMES/AUTO EXPENSES Two long-awaited programs that have got to save you money at work or in the home:
- **Status of Homes**—This program will allow you to keep track of all the expenses involved in building one house or an entire subdivision.
- **Auto Expenses**—Find out *exactly* what it costs you to drive your car or truck.

These programs require a TRS-80 Level I 4K. **Order No. 0012R $7.95.**

PERSONAL FINANCE I Let your TRS-80 handle all the tedious details the next time you figure your finances:
- **Personal Finance I**—With this program you can control your incoming and outgoing expenses.
- **Checkbook**—Your TRS-80 can balance your checkbook and keep a detailed list of expenses for tax time.

This handy financial control for the home requires only a TRS-80 Level I 4K. **Order No. 0027R $7.95.**

SPACE TREK III Let yourself go to the far ends of the solar system—and beyond. This package includes:
- **Stellar Wars**—Shoot down the Tie fighters and destroy the Death Star.
- **Planetary Lander**—Land your spacecraft and plant your flag across the solar system.

These one-player games require a TRS-80 Level I 4K. **Order No. 0031R $7.95.**

DESTROY ALL SUBS/GUNBOATS/BOMBER This package of three programs is fun for the whole family. Included are:
- **Destroy All Subs**—Hunt down enemy subs while avoiding mines and torpedoes. A one-player game.
- **Gunboats**—One or two players can try to blow each other's ships out of the water.
- **Bomber**—Carefully release your bomb to destroy the moving submarine. A one-player game.

To enjoy these programs, you'll need a TRS-80 Level I 4K. **Order No. 0021 R $7.95.**

DEMO I This package is just the thing to show your friends what your TRS-80 can do. Included are:
- **Computer Composer**—Compose and play music using only a standard AM radio.
- **Baseball**—Play baseball with your computer while it does the scorekeeping.
- **Horse Race**—Place your bet and cheer your pony to the winner's circle.
- **ESP**—Test your powers of extrasensory perception.
- **Hi-Lo/Tic-tac-toe**—Guess the secret number or get three in a row.
- **Petals Around the Rose**—Can you figure out the secret behind the five dice?
- **Slot Machine**—Turn your computer into a one-armed bandit. These programs require a TRS-80 Level I 4K. **Order No. 0020R $7.95.**

CAR RACE/RAT TRAP/ANTIAIRCRAFT Enjoy these challenging, fun-filled programs:
- **Car Race**—You and a friend can race on a choice of two tracks.
- **Rat Trap**—Trap the rat in his maze with your two cats. For one player.
- **Antiaircraft**—Aim and shoot down the enemy airplane. Requires Level I 4K TRS-80. **Order No. 0011R $7.95.**

BUSINESS PACKAGE III This package can change your TRS-80 into a full working partner for any businessman:
- **Inventory**—Maintain a computer based inventory for a constant inventory system.
- **Commisions and Percentages**—Let your computer figure out markup and discount calculations, sales tax and more. This is a perfect timesaving package for any small business.

For the TRS-80 Level I 4K. **Order No. 0061R $7.95.**

KNIGHT'S QUEST/ROBOT CHASE/HORSE RACE This varied package of one-player games will give you hours of fun.
- **Knight's Quest**—Battle demons to gain treasure and become a full-fledged knight.
- **Robot Chase**—Destroy the deadly robots without electrocuting yourself.
- **Horse Race**—Place your bet and cheer your horse to the finish line.

These programs require a TRS-80 Level I 16K. **Order No. 0003R $7.95.**

HEX PAWN/SHUTTLE CRAFT DOCKING/SPACE CHASE/SEA BATTLE This four-game package is sure to provide hours of fun for the whole family.
- **Hex Pawn**—Turn your TRS-80 into a model of artificial intelligence by playing a simple game.
- **Shuttle Craft Docking**—Land your shuttle craft on the starship—even through varying gravity fields!
- **Space Chase**—Seek out and destroy the enemy delta that's hidden in the star field.
- **Sea Battle**—You must find and destroy the enemy fleet.

This package requires a TRS-80 Level I 16K. **Order No. 0041R $7.95.**

BUSINESS PACKAGE I Keep the books for a small business with your TRS-80 Level I 4K. The six programs included are:
- **General Information**—The instructions for using the package.
- **Fixed Asset Control**—This will give you a list of your fixed assets and term depreciation.
- **Detail Input**—This program lets you create and record your general ledger on tape for fast access.
- **Month and Year to Date Merge**—This program will take your monthly ledger data and give you a year to date ledger.
- **Profit and Loss**—With this program you can quickly get trial balance and profit and loss statements.
- **Year End Balance**—This program will combine all your data from the profit and loss statements into a year end balance sheet.

With this package, you can make your TRS-80 a working partner. **Order No. 0013R $29.95.**

Level II

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE Buck Rogers never had it so good. Engage in extraterrestrial warfare with:
- **Ramrom Patrol**—Destroy the Ramrom ships before they capture you.
- **Tie Fighter**—Destroy the enemy Tie fighters and become a hero of the rebellion.
- **Klingon Capture**—You must capture the Klingon ship intact. It's you and your TRS-80 Level II 16K battling across the galaxy. **Order No. 0028R $7.95.**

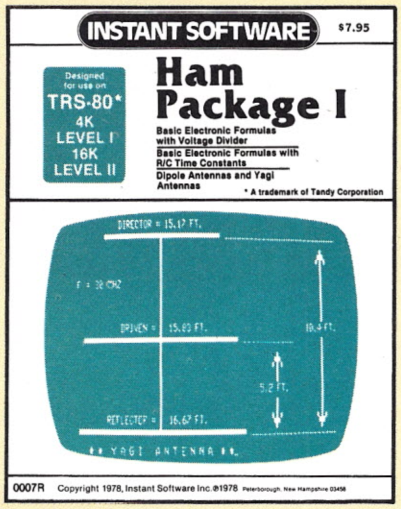
SPACE TREK IV Trade or wage war on a planetary scale. This package includes:
- **Stellar Wars**—Engage and destroy Tie fighters in your attack on the Death Star. For one player.
- **Population Simulation**—A two-player game where you control the economy of two neighboring planets. You decide, guns or butter, with your TRS-80 Level II 16K. **Order No. 0034R $7.95.**

BOWLING LEAGUE STATISTICS SYSTEM This package is the answer to the prayers of harried bowling league scorekeepers. The Bowling League Statistics System will keep a computerized list of league data, team data, and data for each bowler. It is extremely flexible and has a total of 16 different options to let you modify the program to suit your league's rules. The program is very easy to use and has extensive "built in" aids to help you along. Requires TRS-80 Level II 16K. **Order No. 0056R $24.95.**

CARDS This one-player package will let you play cards with your TRS-80—talk about a poker face!
- **Draw and Stud Poker**—These two programs will keep your game sharp.
- **No-Trump Bridge**—Play this popular game with your computer and develop your strategy.

This package's name says it all. Requires a TRS-80 Level II 16K. **Order No. 0063R $7.95.**

Level I and II



HAM PACKAGE I This versatile package lets you solve many of the commonly encountered problems in electronics design. With your Level I 4K or Level II 16K TRS-80, you have a choice of:
- **Basic Electronics with Voltage Divider**—Solve problems involving Ohm's Law, voltage dividers, and RC time constants.
- **Dipole and Yagi Antennas**—Design antennas easily, without tedious calculations.

This is the perfect package for any ham or technician. **Order No. 0007R $7.95.**

BOWLING Let your TRS-80 set up the pins and keep score. One player can pick up spares and get strikes. For the TRS-80 Level I 4K, Level II 16K. **Order No. 0033R $7.95.**

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. This program requires a TRS-80 Level I & II. **Order No. 0043R $7.95.**

AIR FLIGHT SIMULATION Turn your TRS-80 into an airplane. You can practice takeoffs and landings with the benefit of full instrumentation. This one-player simulation requires a TRS-80 Level I 4K, Level II 16K. **Order No. 0017R $7.95.**

GOLF/CROSSOUT Have fun with these exciting one-player games. Included are:
- **Golf**—You won't need a mashie or putter—or a caddie, for that matter, to enjoy a challenging 18 holes.
- **Crossout**—Remove all but the center peg in this puzzle and your neighbors will call you a genius.

You'll need a TRS-80 Level I 4K, Level II 16K. **Order No. 0009R $7.95.**

BASIC AND INTERMEDIATE LUNAR LANDER Bring your lander in under manual control. The Basic version is for beginners; the Intermediate version is more difficult with a choice of landing areas and rugged terrain. For one player with a TRS-80 Level I 4K, Level II 16K. **Order No. 0001R $7.95.**

SPACE TREK II Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power, and warp drive. It's you alone and your TRS-80 Level I 4K, Level II 16K against the enemy. **Order No. 0002R $7.95.**

CONOPS

This control operating system is a monitor for the Heath H8.

Chesney E. Twombly
15 Storer Street
Kennebunk ME 04043

This is a monitor for a Heath H8-video terminal-cassette I/O system. It enables you to talk with your computer in hexadecimal, makes it easy to load and run any 8080 assembly-language program and makes maximum use of the ROM panel monitor, which contains the reliable, 1200-baud cassette program. The console driver software is used and must be in RAM. No hardware modifications are needed.

H8 users who want to load machine-language programs from a video terminal may be surprised to learn that BUG-8, the console debugger, is not much help—in fact, you may as well use the front-panel keypad. Considering Heath's otherwise excellent H8 software, the lack of a console-oriented operating system is hard to justify.

Since my interest was mostly self-instruction in assembly-language programming, I thought it worthwhile to develop an operating system to expedite this kind of activity. The console operating system, which I will refer to as CONOPS, is presented here in an assembly-language listing. A collection of several of the more useful routines I have used over the past year, it has been thoroughly tested and can be modified easily to satisfy individual preferences.

The listing is in octal to enable a prospective user to load the program from the H8 front panel. CONOPS uses hexadecimal exclusively, octal being needed only for the initial jump to 6C00, the starting address. Although Heath software does not recognize hexadecimal, it is my preference for two reasons: (1) saves one keystroke per byte when I load programs from a keyboard; (2) simple conversion to binary. Arithmetic follows familiar rules.

For ultimate efficiency in dealing with hexadecimal, I recommend the TI Programmer, Texas Instruments' unique calculator that works in three number systems and is extremely handy for making conversions between any two of them. It is available from Heath Company and probably from many of the computer stores. Even the TI Programmer, however, cannot directly add or subtract two addresses in split-octal.

Comparing different number systems is irrelevant if you use only BASIC or canned, machine-language programs. The CPU has to have it in binary and doesn't care how it gets there.

Memory Considerations

The best location for CONOPS is as high in memory as possible. Remember that the highest 80-byte block of continuous RAM must be reserved for the H8 panel monitor. According to Heath convention, the lowest RAM address should be at 040.000 or 2000 hex.

I started with an 8K memory board from Heath and a 12K board from Bill Godbout, which gave me addressable memory from 2000-6FFF hex. CONOPS uses only 788 bytes and was assembled to go between 6C00 and 6F14, with a stack initialized at 6FB0. You can relocate it quite easily to fit whatever range of RAM you happen to have, but remember to change all direct addresses that are in the block that is moved.

The console driver can be loaded from any H8 software tape. It is on all of them right at the end that loads first, and you can get it all in about 10 seconds. When you're not using Heath software, it is a good idea to change the contents of 2144 to C9. This disables control-character processing by the console driver, a problem if your program uses control characters. You can enter the jump command, C3 00 6C, at 2040, so you go directly to CONOPS upon hitting the H8 GO button.

```
AA     154.020  ABUSS  040.024  ADDOUT 154.170  ADDS   154.173  AMOR   156.132
ANO    155.200  BBB    154.350  BLKMOV 155.075  BYTE   154.100  CBEG   001.374
CEND   002.062  CHECK  155.161  CHKPC  156.072  CMPDH  154.361  CONVT  154.050
CRCSUM 040.027  CRFUNC 154.021  CRLF   154.032  DD     155.152  DUMP   154.372
FF     155.232  GETADR 156.006  GETCH  155.011  GETCHR 154.233  GOTO   155.345
HEXL   154.125  HEXR   154.131  HORN   002.140  IN4H   154.114  INBUF  157.104
INBYTE 154.075  INCHR  154.035  INHEX  154.044  LOOP   155.047  LU2    155.062
LUP    155.366  MEMORY 046.316  MONIT  156.333  MORE   155.003  MOV1   155.135
MS1    155.264  MS2    155.303  MS3    155.322  MS4    155.336  MS5    155.274
MWRITE 156.204  NEWST  154.272  OK     154.067  OPTPC  000.371  OUT    155.072
OUT2H  154.142  OUT2HS 154.160  OUTCH  040.147  OUTS   154.206  PADDR  154.247
PC     157.102  PDATA  154.003  PGMLDR 154.214  PRSCL  040.152  RCHAR  040.144
READ   001.261  STACK  157.260  START  156.301  STOP   154.262  TEMP   157.100
TPABT  002.244  TPERRX 040.031  TSTAR  040.000  WMEX   002.074  WNB    003.024
WNP    003.017  WRITE  155.375  ZBUF   155.360  ZMEM   154.325
```

Symbol table.

After correcting any loading errors, you can use the Write function to save CONOPS on a cassette, along with the console driver. Blocks 6C00-6F14 and 201F-2163 and starting address 2040 can be entered at one time; the two blocks will be recorded as successive files with a 5-second gap separating them. You have to start and stop the tape drive manually, and the changing number display on the H8 panel indicates that recording is in progress. An audio beep will tell you when both blocks have been recorded.

The number of blocks you can enter serially is determined by the length of the address buffer, which starts at 6F40. Four bytes of buffer space is needed for storage of the starting and ending addresses of each block. The buffer will hold ten blocks plus a starting address. I have never used more than five, but you can extend the buffer if you want to. The subroutine ZBUF is used to clear the buffer, which has the starting address TEMP. The buffer has to be cleared because the program looks for a zero byte to recognize that the final block has been recorded.

Functions and Formats

You have loaded CONOPS and the console driver; now comes the fun part—finding out what it will do.

Use the H8 keypad to set the PC to a starting address of 154.000, hit the GO button and you should see the prompt, ***, on your video-terminal screen. This tells you that CONOPS is waiting for a command. The Command Table section of the program listing shows the program selection codes. Each code is a single letter, easily memorized because it is the first letter of the program name.

Program Loader (P). Push the P key to activate the program loader. You will see the following on the screen: BEGIN ADDR? Now, enter the address where you wish to store the first byte of the program being loaded. This address must consist of four hexadecimal digits —anything else will be rejected. As an example, enter 6000 and start entering CONOPS itself. After some data has been loaded, your video terminal should show you this:

```
*** P
BEGIN ADDR? 6000
6000 C3 C1 6E 7E FE 04 CA 10
6008 6C CD 67 20 23 C3 03 6C
6010 C9 S
***
```

Each pair of characters represents eight binary digits. The following command options are recognized and any of them can be keyed in after entry of a complete byte:

L—will cause the next address to be displayed.

<—cancels the last byte entered.

R—restarts loader with a new "BEGIN ADDR?" prompt.

S—gets you out of the loader and back to ***.

Zero Memory (Z). A Z will call up the zero-memory routine. The prompts appear as follows:

```
*** Z
BEGIN ADDR? 6000
  END ADDR? 6010
***
```

Enter the begin and end addresses of the block to be cleared. The example above indicates that 6000-6010 will be erased. The program tests to verify that the end address is higher than the begin. If not, the program will return to *** without clearing any memory. Memory blocks of moderate length

Program lisiting

```
MONITOR A
PASS = 2
                        * A USEFUL H8 OPERATING SYSTEM
                        *
                        *    BY CHESNEY E. TWOMBLY
                        *       15 STORER ST.,
                        *     KENNEBUNK ME 04043
                        *
                                OPT     NUM,MEM
                                OPT     OCT
    154.000                     ORG     6C00H
                        *
                157.260 STACK   EQU     6FB0H
                040.147 OUTCH   EQU     2067H
                040.144 RCHAR   EQU     2064H
                040.152 PRSCL   EQU     206AH
                001.261 READ    EQU     01B1H
                040.000 TSTAR   EQU     2000H
                040.024 ABUSS   EQU     2014H
                157.104 INBUF   EQU     6F44H
                157.102 PC      EQU     6F42H
                157.100 TEMP    EQU     6F40H
                046.316 MEMORY  EQU     26CEH
                002.140 HORN    EQU     0260H
                040.031 TPERRX  EQU     2019H
                003.024 WNB     EQU     0314H
                040.027 CRCSUM  EQU     2017H
                003.017 WNP     EQU     030FH
                002.244 TPABT   EQU     02A4H
                000.371 OPTPC   EQU     0F9H
                002.074 WMEX    EQU     023CH
                001.374 CBEG    EQU     01FCH
                002.062 CEND    EQU     0232H
                        *
    154.000 303 301 156         JMP     START
                        *
    154.003 176         PDATA   MOV     A,M
    154.004 376 004             CPI     4
    154.006 312 020 154         JZ      AA
    154.011 315 147 040         CALL    OUTCH
    154.014 043                 INX     H
    154.015 303 003 154         JMP     PDATA
    154.020 311         AA      RET
                        *
    154.021 345         CRFUNC  PUSH    H
    154.022 041 032 154         LXI     H,CRLF
    154.025 315 003 154         CALL    PDATA
    154.030 341                 POP     H
    154.031 311                 RET
                        *
    154.032 015 012 004 CRLF    DB      0DH,0AH,4
                        *
    154.035 315 144 040 INCHR   CALL    RCHAR
    154.040 315 147 040         CALL    OUTCH
    154.043 311                 RET
                        *
    154.044 247         INHEX   ANA     A
    154.045 315 035 154         CALL    INCHR
    154.050 326 107     CONVT   SUI     47H
    154.052 362 044 154         JP      INHEX
    154.055 306 006             ADI     6
    154.057 362 067 154         JP      OK
    154.062 306 007             ADI     7
    154.064 362 044 154         JP      INHEX
    154.067 306 012     OK      ADI     0AH
    154.071 372 044 154         JM      INHEX
    154.074 311                 RET
                        *
    154.075 315 044 154 INBYTE  CALL    INHEX
    154.100 007         BYTE    RLC
    154.101 007                 RLC
    154.102 007                 RLC
    154.103 007                 RLC
    154.104 305                 PUSH    B
    154.105 107                 MOV     B,A
    154.106 315 044 154         CALL    INHEX
    154.111 200                 ADD     B
    154.112 301                 POP     B
    154.113 311                 RET
                        *
    154.114 315 075 154 IN4H    CALL    INBYTE
    154.117 127                 MOV     D,A
    154.120 315 075 154         CALL    INBYTE
    154.123 137                 MOV     E,A
    154.124 311                 RET
                        *
    154.125 017         HEXL    RRC
    154.126 017                 RRC
    154.127 017                 RRC
    154.130 017                 RRC
    154.131 346 017     HEXR    ANI     0FH
    154.133 306 220             ADI     90H
    154.135 047                 DAA
    154.136 316 100             ACI     40H
    154.140 047                 DAA
    154.141 311                 RET
                        *
    154.142 176         OUT2H   MOV     A,M
    154.143 315 125 154         CALL    HEXL
    154.146 315 147 040         CALL    OUTCH
    154.151 176                 MOV     A,M
    154.152 315 131 154         CALL    HEXR
    154.155 303 147 040         JMP     OUTCH
    154.160 315 142 154 OUT2HS  CALL    OUT2H
    154.163 076 040             MVI     A,20H
    154.165 303 147 040         JMP     OUTCH
                        *
    154.170 315 021 154 ADDOUT  CALL    CRFUNC
```

```
154.173 Ø41 1ØØ 157  ADDS    LXI    H,TEMP
154.176 162                  MOV    M,D
154.177 315 142 154          CALL   OUT2H
154.2Ø2 163                  MOV    M,E
154.2Ø3 315 142 154          CALL   OUT2H
154.2Ø6 Ø76 Ø4Ø      OUTS    MVI    A,2ØH
154.21Ø 315 147 Ø4Ø          CALL   OUTCH
154.213 311                  RET
                     *
                     ***  PROGRAM LOADER  ***
                     *
154.214 Ø61 26Ø 157  PGMLDR  LXI    SP,STACK
154.217 Ø41 264 155          LXI    H,MS1
154.222 315 ØØ3 154          CALL   PDATA
154.225 315 114 154          CALL   IN4H
154.23Ø 315 17Ø 154          CALL   ADDOUT
154.233 315 Ø35 154  GETCHR  CALL   INCHR
154.236 376 Ø74              CPI    '<'           ;DELETE LAST CHAR
154.24Ø 3Ø2 247 154          JNZ    PADDR
154.243 Ø33                  DCX    D
154.244 3Ø3 233 154          JMP    GETCHR
154.247 376 114      PADDR   CPI    'L'           ;LIST NEXT ADDR
154.251 3Ø2 262 154          JNZ    STOP
154.254 315 17Ø 154          CALL   ADDOUT
154.257 3Ø3 233 154          JMP    GETCHR
154.262 376 123      STOP    CPI    'S'           ;ALL DONE
154.264 3Ø2 272 154          JNZ    NEWST         ;NEW START
154.267 3Ø3 333 156          JMP    MONIT
154.272 376 122      NEWST   CPI    'R'
154.274 312 214 154          JZ     PGMLDR
154.277 3Ø4 Ø5Ø 154          CNZ    CONVT
154.3Ø2 315 1ØØ 154          CALL   BYTE
154.3Ø5 Ø22                  STAX   D
154.3Ø6 315 2Ø6 154          CALL   OUTS
154.311 Ø23                  INX    D
154.312 173                  MOV    A,E
154.313 346 ØØ7              ANI    7
154.315 376 ØØØ              CPI    Ø
154.317 314 17Ø 154          CZ     ADDOUT
154.322 3Ø3 233 154          JMP    GETCHR
                     *
                     ***  ZERO MEMORY  ***
                     *
154.325 Ø41 264 155  ZMEM    LXI    H,MS1
154.33Ø 315 ØØ3 154          CALL   PDATA
154.333 315 114 154          CALL   IN4H
154.336 Ø41 3Ø3 155          LXI    H,MS2
154.341 315 ØØ3 154          CALL   PDATA
154.344 353                  XCHG
154.345 315 114 154          CALL   IN4H
154.35Ø Ø66 ØØØ      BBB     MVI    M,Ø
154.352 315 361 154          CALL   CMPDH
154.355 Ø43                  INX    H
154.356 3Ø3 35Ø 154          JMP    BBB
154.361 172          CMPDH   MOV    A,D
154.362 254                  XRA    H
154.363 3ØØ                  RNZ
154.364 173                  MOV    A,E
154.365 255                  XRA    L
154.366 3ØØ                  RNZ
154.367 3Ø3 333 156          JMP    MONIT
                     *
                     ***  DUMP MEMORY  ***
                     *
154.372 Ø41 264 155  DUMP    LXI    H,MS1
154.375 315 ØØ3 154          CALL   PDATA
155.ØØØ 315 114 154          CALL   IN4H
155.ØØ3 315 Ø21 154  MORE    CALL   CRFUNC
155.ØØ6 315 173 154          CALL   ADDS
155.Ø11 Ø32          GETCH   LDAX   D
155.Ø12 Ø41 1ØØ 157          LXI    H,TEMP
155.Ø15 167                  MOV    M,A
155.Ø16 315 16Ø 154          CALL   OUT2HS
155.Ø21 Ø23                  INX    D
155.Ø22 173                  MOV    A,E
155.Ø23 376 2ØØ              CPI    8ØH
155.Ø25 312 Ø47 155          JZ     LOOP
155.Ø3Ø 376 ØØØ              CPI    Ø
155.Ø32 312 Ø47 155          JZ     LOOP
155.Ø35 346 Ø17              ANI    ØFH
155.Ø37 376 ØØØ              CPI    Ø
155.Ø41 314 17Ø 154          CZ     ADDOUT
155.Ø44 3Ø3 Ø11 155          JMP    GETCH
155.Ø47 315 Ø35 154  LOOP    CALL   INCHR
155.Ø52 376 1Ø6              CPI    'F'
155.Ø54 3Ø2 Ø62 155          JNZ    LU2
155.Ø57 3Ø3 ØØ3 155          JMP    MORE
155.Ø62 376 122      LU2     CPI    'R'
155.Ø64 3Ø2 Ø72 155          JNZ    OUT
155.Ø67 3Ø3 372 154          JMP    DUMP
155.Ø72 3Ø3 333 156  OUT     JMP    MONIT
                     *
                     ***  BLOCK MOVE  ***
                     *
155.Ø75 Ø41 322 155  BLKMOV  LXI    H,MS3
155.1ØØ 315 ØØ3 154          CALL   PDATA
155.1Ø3 315 114 154          CALL   IN4H
155.1Ø6 325                  PUSH   D
155.1Ø7 Ø41 264 155          LXI    H,MS1
155.112 315 ØØ3 154          CALL   PDATA
155.115 315 114 154          CALL   IN4H
155.12Ø 325                  PUSH   D
155.121 Ø41 3Ø3 155          LXI    H,MS2
```

```
*** D
BEGIN ADDR? 6C00
6C00 C3 C1 6E 7E FE 04 CA 10 6C CD 67 20 23 C3 03 6C
6C10 C9 E5 21 1A 6C CD 03 6C E1 C9 0D 0A 04 CD 64 20
6C20........
.............
6C70 CD 62 6C 3E 20 C3 67 20 CD 11 6C 21 40 6F 72 CD S
***
```

Example 1.

are cleared so quickly that you may think nothing has happened, but you can use the dump-memory routine, described next, to verify that clearing has been done.

Dump Memory (D). This routine gives a console display of memory contents, starting at an address entered from the keyboard. Example 1 illustrates the format. The display will stop upon reaching an address ending in 7F or FF. At that point, the sequence can be continued by entering an F command. An S will return the program to ***. If the starting address ends in 00 or 80, the block displayed will be 128 decimal bytes long.

Block Move (B). Moves a block of memory to a location starting at an address entered by the operator. The following example shows the format and indicates a move of block 6C00-6C7F to address 6000:

```
*** B
MOVE TO? 6000
BEGIN ADDR? 6C00
  END ADDR? 6C7F
***
```

Check and Modify Memory (C). Used to check the contents of any location in memory, with the option of changing it. Hitting the space bar will display the next higher address, and an S will return the program to ***. To modify a value, type a / and follow with entry of the new byte. The following format will be seen on your terminal:

```
*** C
6000 C3
6001 C1
6002 6E/FF
6003 7E
6004 S
***
```

GOTO and Execute (G). Transfers control from CONOPS to a program in user memory space. Enter G and the starting address of the user program. There will be no return to CONOPS unless the program contains an instruction sequence to cause a return.

Write a File to Cassette (W). Records from one to ten independent data blocks, of any length, on cassette tape, in H8 Memory Image format, using all the bells and whistles you are accustomed to from using the panel monitor. The operator enters the start and ending addresses of each block, using a comma to separate the blocks.

When you finish entering the block addresses, hit the colon and follow with entry of the address assignment for the program counter, usually 2040. With the tape running, hit the space bar to start recording. The blocks will be recorded in the sequence entered, with a separation of about five seconds. The console display will be similar to the following example:

```
*** W
6C00 6F14
201F 2163:2040
***
```

Read Tape (R). Reads a cassette tape in H8 Memory Image format into RAM. Start the tape and hit R just before the start of each data file to be read. Control returns to CONOPS.

Final Remarks

The listing accompanying this article was assembled by a TSC 8080 Mnemonic Assembler from text prepared on a Scelbi 8080 Standard Editor, modified for line numbering. The Editor/Assembler programs are co-resident in memory and share a common text buffer—a very nice arrangement. The foregoing is an example of what you can do with your versatile H8 with the help of CONOPS, a useful operating system. ■

```
155.124 315 003 154          CALL   PDATA
155.127 315 114 154          CALL   IN4H
155.132 353                  XCHG
155.133 301                  POP    B
155.134 321                  POP    D
155.135 170         MOV1     MOV    A,B
155.136 274                  CMP    H
155.137 302 152 155          JNZ    DD
155.142 171                  MOV    A,C
155.143 275                  CMP    L
155.144 302 152 155          JNZ    DD
155.147 012                  LDAX   B
155.150 022                  STAX   D
155.151 311                  RET
155.152 012         DD       LDAX   B
155.153 022                  STAX   D
155.154 003                  INX    B
155.155 023                  INX    D
155.156 303 135 155          JMP    MOV1
                    *
                    ***  CHECK AND MODIFY MEMORY  ***
                    *
155.161 315 021 154 CHECK    CALL   CRFUNC
155.164 041 274 155          LXI    H,MS5
155.167 315 003 154          CALL   PDATA
155.172 315 114 154          CALL   IN4H
155.175 315 021 154          CALL   CRFUNC
155.200 315 173 154 ANO      CALL   ADDS
155.203 032                  LDAX   D
155.204 041 100 157          LXI    H,TEMP
155.207 167                  MOV    M,A
155.210 315 142 154          CALL   OUT2H
155.213 315 144 040          CALL   RCHAR
155.216 376 040              CPI    20H
155.220 302 232 155          JNZ    FF
155.223 315 021 154          CALL   CRFUNC
155.226 023                  INX    D
155.227 303 200 155          JMP    ANO
155.232 376 123     FF       CPI    'S'
155.234 312 333 156          JZ     MONIT
155.237 376 057              CPI    '/'
155.241 302 161 155          JNZ    CHECK
155.244 076 057              MVI    A,2FH
155.246 315 147 040          CALL   OUTCH
155.251 315 075 154          CALL   INBYTE
155.254 022                  STAX   D
155.255 023                  INX    D
155.256 315 021 154          CALL   CRFUNC
155.261 303 200 155          JMP    ANO
                    *
                    ***  MESSAGE LIST  ***
                    *
155.264 015 012     MS1      DB     0DH,0AH
155.266 102 105 107          DB     'BEGIN '
155.271 111 116 040
155.274 101 104 104 MS5      DB     'ADDR? ',4
155.277 122 077 040
155.302 004
155.303 015 012     MS2      DB     0DH,0AH
155.305 040 040 105          DB     '  END ADDR? ',4
155.310 116 104 040
155.313 101 104 104
155.316 122 077 040
155.321 004
155.322 015 012     MS3      DB     0DH,0AH
155.324 115 117 126          DB     'MOVE TO? ',4
155.327 105 040 124
155.332 117 077 040
155.335 004
155.336 015 012     MS4      DB     0DH,0AH
155.340 052 052 052          DB     '*** ',4
155.343 040 004
                    *
                    ***  GO TO AND EXECUTE PROGRAM  ***
                    *
155.345 315 206 154 GOTO     CALL   OUTS
155.350 315 114 154          CALL   IN4H
155.353 315 021 154          CALL   CRFUNC
155.356 353                  XCHG
155.357 351                  PCHL
                    *
155.360 257         ZBUF     XRA    A
155.361 021 100 157          LXI    D,TEMP
155.364 006 057              MVI    B,2FH
155.366 022         LUP      STAX   D
155.367 023                  INX    D
155.370 005                  DCR    B
155.371 302 366 155          JNZ    LUP
155.374 311                  RET
                    *
                    ***  WRITE A MEMORY IMAGE FILE TO CASSETTE  ***
                    *
155.375 315 021 154 WRITE    CALL   CRFUNC
156.000 315 360 155          CALL   ZBUF
156.003 041 104 157          LXI    H,INBUF
156.006 315 114 154 GETADR   CALL   IN4H
156.011 325                  PUSH   D
156.012 163                  MOV    M,E
156.013 043                  INX    H
156.014 162                  MOV    M,D
156.015 076 040              MVI    A,20H
156.017 315 147 040          CALL   OUTCH
156.022 043                  INX    H
156.023 315 114 154          CALL   IN4H
156.026 163                  MOV    M,E
156.027 043                  INX    H
156.030 162                  MOV    M,D
156.031 043                  INX    H
156.032 042 100 157          SHLD   TEMP
156.035 353                  XCHG
156.036 321                  POP    D
156.037 247                  ANA    A
156.040 175                  MOV    A,L
156.041 223                  SUB    E
156.042 157                  MOV    L,A
156.043 174                  MOV    A,H
156.044 232                  SBB    D
156.045 147                  MOV    H,A
156.046 332 333 156          JC     MONIT
156.051 315 035 154          CALL   INCHR
156.054 376 054              CPI    ','
156.056 302 072 156          JNZ    CHKPC
156.061 315 021 154          CALL   CRFUNC
156.064 052 100 157          LHLD   TEMP
156.067 303 006 156          JMP    GETADR
156.072 376 072     CHKPC    CPI    ':'
156.074 302 333 156          JNZ    MONIT
156.077 315 114 154          CALL   IN4H
156.102 353                  XCHG
156.103 042 102 157          SHLD   PC
156.106 076 017              MVI    A,0FH
156.110 315 140 002          CALL   HORN
156.113 315 035 154          CALL   INCHR
156.116 376 040              CPI    '
156.120 302 333 156          JNZ    MONIT
156.123 315 021 154          CALL   CRFUNC
156.126 041 104 157          LXI    H,INBUF
156.131 176                  MOV    A,M
156.132 062 000 040 AMOR     STA    TSTAR
156.135 043                  INX    H
156.136 176                  MOV    A,M
156.137 062 001 040          STA    TSTAR+1
156.142 043                  INX    H
156.143 176                  MOV    A,M
156.144 062 024 040          STA    ABUSS
156.147 043                  INX    H
156.150 176                  MOV    A,M
156.151 062 025 040          STA    ABUSS+1
156.154 043                  INX    H
156.155 345                  PUSH   H
156.156 315 204 156          CALL   MWRITE
156.161 341                  POP    H
156.162 176                  MOV    A,M
156.163 376 000              CPI    0
156.165 302 132 156          JNZ    AMOR
156.170 043                  INX    H
156.171 176                  MOV    A,M
156.172 376 000              CPI    0
156.174 312 333 156          JZ     MONIT
156.177 053                  DCX    H
156.200 176                  MOV    A,M
156.201 303 132 156          JMP    AMOR
156.204             MWRITE   DS     36H
156.272 325                  PUSH   D
156.273 052 102 157          LHLD   PC
156.276 303 074 002          JMP    WMEX
                    *
                    ***  COMMAND TABLE  ***
                    *
156.301 315 152 040 START    CALL   PRSCL
156.304 061 260 157          LXI    SP,STACK
156.307 001 374 001          LXI    B,CBEG
156.312 041 204 156          LXI    H,MWRITE
156.315 353                  XCHG
156.316 041 062 002          LXI    H,CEND
156.321 053                  DCX    H
156.322 315 135 155          CALL   MOV1
156.325 041 222 156          LXI    H,MWRITE+0EH
156.330 042 227 156          SHLD   MWRITE+13H
156.333 061 260 157 MONIT    LXI    SP,STACK
156.336 041 336 155          LXI    H,MS4
156.341 315 003 154          CALL   PDATA
156.344 315 035 154          CALL   INCHR
156.347 376 102              CPI    'B'
156.351 314 075 155          CZ     BLKMOV
156.354 376 104              CPI    'D'
156.356 312 372 154          JZ     DUMP
156.361 376 120              CPI    'P'
156.363 312 214 154          JZ     PGMLDR
156.366 376 132              CPI    'Z'
156.370 312 325 154          JZ     ZMEM
156.373 376 103              CPI    'C'
156.375 312 161 155          JZ     CHECK
157.000 376 122              CPI    'R'
157.002 314 261 001          CZ     READ
157.005 376 127              CPI    'W'
157.007 312 375 155          JZ     WRITE
157.012 376 107              CPI    'G'
157.014 314 345 155          CZ     GOTO
157.017 376 110              CPI    'H'
157.021 302 333 156          JNZ    MONIT
157.024 166                  HLT
                    *
                             END
```

Getting the Most out of Your TRS-80

This article deals particularly with deriving the most from the 80's graphics capabilities.

William L. Colsher
4328 Nutmeg Lane, Apt. 111
Lisle IL 60532

You can do a lot with the graphic capability that Radio Shack has built into the TRS-80 as it comes from the factory. If you're willing to invest $14.95 in a copy of T-BUG, though, you can open up a whole new world of possibilities. In addition to the standard character set that the TRS-80 can display, there are also 64 more graphics characters hidden in the character generator. In this article we'll take a quick look at what they are and how to make use of them.

The screen of the TRS-80 is divided up into a 48 × 128 rectangular grid. Each of these rectangles can be turned on or off by a BASIC command. Ordinary characters take up six of these tiny blocks in the form of a two-wide by three-high rectangle. In addition to these ordinary characters, there are 64 more characters: one for each possible combination of those six rectangles. There is no duplication of the regular characters, since they completely ignore the block boundaries.

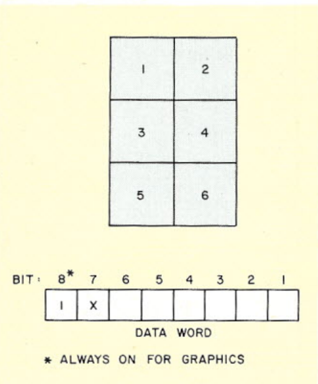


Fig. 1. Character-generation block numbering.

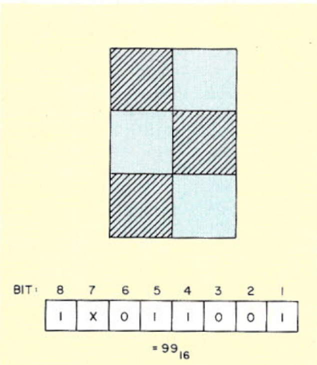


Fig. 2. Sample block graphic.

Radio Shack didn't just haphazardly throw the characters together; there is a definite pattern. To even get a graphic character, the high-order bit of the word must be on. This puts us into the right part of the character generator. Since there are six little rectangles in a block, it makes sense to just use one bit to indicate the status (on or off) of each of the little rectangles. This is exactly what they've done.

The rectangles are numbered from left to right and top to bottom, 1 through 6 (see Fig. 1). The number of a given rectangle is the number of its corresponding bit. Fig. 2 shows an example, and Fig. 3 shows all the available combinations.

The 64 graphic characters don't look like too much all by themselves, as you can see from Fig. 3. But by using more than one, you can form interesting combinations. For example, the two characters 99H and A6H placed next to each other make a realistic Tie-Fighter for a Star Wars game. This combination and several others are shown in Fig. 4.

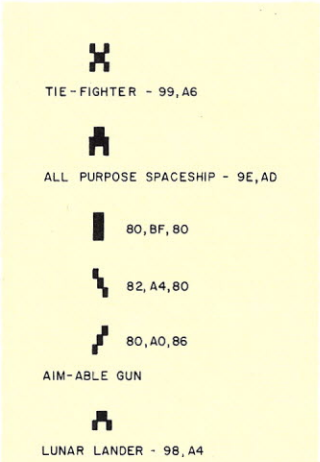


Fig. 4. Sample character combinations.

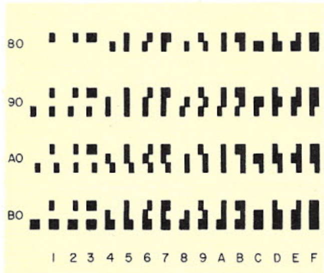
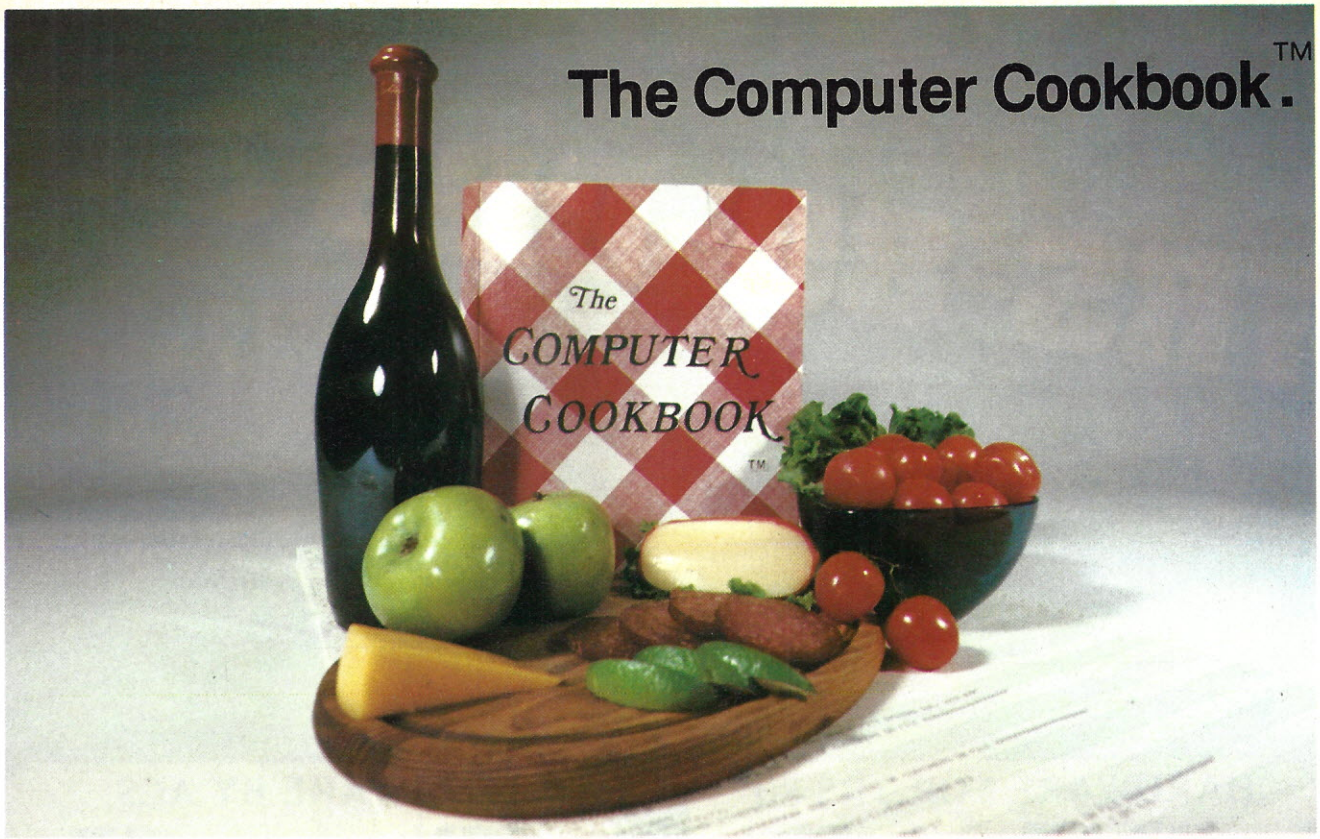


Fig. 3. TRS-80 graphics characters.

Of course, some people will feel that having to use T-BUG and machine language is a disadvantage. (You should be able to do this with Level II BASIC also.) However, there's one advantage of machine-language programming: speed.

For example, you can fill the entire screen with the character of your choice so quickly that it seems the characters were all written at the same time. This kind of performance enables you to bring about some clever animation.

This little article should give you some ideas for future programs. ■



The Only Computer Reference Book You Need. The Only Reference Book You'll Ever Need.

The only book that gives you all the ingredients for a successful computer system. Interested in the Apple II?™ You'll need an FCC-approved RF modulator to connect it to your color television set. We tell what they are, where to get one, and how to hook it up.

Thinking of using Radio Shack's TRS-80™ for a low-cost word processing system? You'll need an upper-case/lower case modification for the keyboard. We'll tell you what it involves, where to get it done, or how to do it yourself.

Software? We've sampled canned software and will tell you what suits our taste. Rather do it yourself? We have algorithms in our machine-independent "Cookbook" language for everything from fast sorts to phototypesetter hyphenation. Plus a crash course in BASIC for those who know FORTRAN. And much, much more.

Last but not least: The Cookbook Yellow Pages, a complete guide to small computer services and products. We update The Yellow Pages every two months—and send them to you free with the latest additions, corrections, and improvements of our white pages. You just plug them into our loose-leaf binder.

Get with The Cookbook. The Best is Yet to Come.

Send me The Cookbook. I enclose $15 plus $1.50 for shipping, which will be by UPS unless I've marked this form U.S. mail (____check here). California residents add $.98 sales tax. Allow up to four weeks for possible out-of-stock conditions. Make checks payable to "The Computer Cookbook". Mastercharge and VISA accepted with 13-digit card number and 4-digit expiration date.

Name ____________________

Street ____________________

City ____________________ State ____________ Zip ____________

Mail to: The Computer Cookbook / P.O. Box 4084 / Berkeley, CA 94704 C141

David Price
3901 Victoria Lane
Midlothian VA 23113

Reading Computer Jargon

The words that have evolved in microcomputing are concise, but what do they mean?

The subset of the English language that we refer to as "computer jargon" was not created spontaneously. Rather, it evolved as the industry evolved, with new words and phrases being added as they were needed. The rich lingo of the computer field permits the expression of complex ideas with extraordinary ease and concision.

Unfortunately, the newcomer who is not familiar with this jargon misses a great deal, because he cannot digest much of the information contained in computer magazine articles, computer advertisements and other sources. This glossary represents my attempt at explaining some of the more common terms. They are arranged into four categories: central processing unit, input/output devices, software and miscellaneous.

Central Processing Unit

Access time: The amount of time required to access the contents of a memory location. This limitation is imposed by the speed of the memory circuitry.

ALU (arithmetic logic unit): The portion of the central processor that executes all mathematical and logical functions.

Architecture: The internal configuration of a processor including its registers and instruction set.

Backplane: The circuitry and mechanical elements used to connect the boards of a system. See also *motherboard* and *card frame.*

Bit: A single binary digit. A bit may have two states: normally, a bit is considered "high" when its value is one and "low" when its value is zero.

Board: A card that contains circuitry for one or more specific functions, such as memory or interfacing.

Bus: The circuitry in a backplane that allows transmission of electrical signals from one board to another.

Bus structure: The definition of a bus in terms of its mechanical requirements and the functions of its lines. A memory board designed for the S-100 bus, for example, would not be compatible with a computer using the IEEE-488 bus or the SS-50 bus.

Byte: A group of eight bits that are treated as a unit. The values of the bits can be varied to form as many as 2^8 or 256 permutations. Hence, one byte of memory can represent an integer from 0 to 255 or from -127 to $+128$.

Card frame: An enclosure that holds a system's boards in place.

Clock: A device that generates electronic timing signals. Clock signals are often used to synchronize certain system operations.

Core memory: A type of memory that stores information on magnetically charged, doughnut-shaped cores made of ferrite and lithium. Core memories have largely been superseded by semiconductor memories.

CPU (central processing unit): The primary component of all computer systems. It is responsible for controlling system operation as directed by the program it is executing.

Cycle Stealing: A technique that allows an external device to temporarily disable processor control of the bus. This, in turn, allows the device to access main memory.

DMA (direct memory access): An arrangement where blocks of data can be transferred between main memory and a peripheral device (such as a disk drive) without processor intervention.

Dynamic memory: A type of semiconductor memory which, unlike static memory, must be refreshed or recharged periodically to prevent loss of data.

EAROM (electrically alterable read-only memory): A type of memory that combines the characteristics of RAM and read-only memory. It is nonvolatile (like read-only memory) but can be written into by the processor (like RAM). The EAROM, however, has a substantially longer writing time (currently about 2 microseconds vs 400 nanoseconds) as well as a limited number of writes (about 1,000,000) before the chip can no longer be reprogrammed.

EROM (erasable read-only memory): A read-only memory that can be erased and reprogrammed. Most EROMs are erased through exposure to ultraviolet light. Also spelled EPROM.

Extender board: A troubleshooting aid. It physically raises another board above the other boards in a system, where it can be monitored more conveniently.

Firmware: Software stored in read-only memory. Also a synonym for microcode.

Hardware multiply/divide: A feature that allows a processor to perform multiplication and division entirely within hardware. Processors without multiply and divide instructions require special software routines for these operations.

Invisible refresh: A scheme that refreshes dynamic memories without disturbing the rest of the system. The refresh requirements of the memories might otherwise reduce system performance and interfere with DMA operations.

Mainframe: The computer itself, including the processor, main memory, I/O interfaces and backplane.

Main memory: The memory that

the processor accesses directly, as opposed to peripherals such as disk and tape devices.

Mask ROM: A read-only memory that is permanently programmed by the chip manufacturer.

Motherboard: Synonym for backplane.

MPU (microprocessor unit).

ns (nanosecond): A billionth (10^{-9}) of a second.

On-board regulation: An arrangement where each board in a system contains its own voltage regulator.

Processor: Synonym for CPU.

Programmable memory: Memory that can be both read from and written into by the processor. Synonym for RAM.

PROM (programmable read-only memory): A type of read-only memory that can be programmed by the user. This programming usually requires special equipment. See *burning* under Input/Output Devices.

Protected memory: Programmable memory that cannot be written into, usually on a temporary basis.

RAM (random access memory): See *programmable memory.*

Real-Time clock: An electronic timekeeping device within the computer.

ROM (read-only memory): Memory whose preprogrammed contents cannot be altered by the processor.

Scratchpad memory: Programmable memory that is being used for storage of "housekeeping" value, or information that is internal to the program.

Semiconductor memory: Memory consisting of integrated circuits rather than magnetic cores.

Slot: A single-board position in a backplane.

Static memory: Memory that does not require refreshing.

Throughput: A loosely defined term that refers to the speed of a processor.

Turnkey: A computer whose front panel is blank or contains few controls. Also refers to a product delivered ready to run.

Volatile memory: Memory that loses its contents when operating power is removed.

Word: A basic unit of computer memory. The length of the word may vary from processor to processor. The most common microcomputer word length is 8 bits, or one byte.

Input/Output Devices

ACIA (asynchronous communications interface adapter): See *UART.*

Acoustic coupler: A device that allows digital information to be transmitted over voice-grade telephone lines.

A/D (analog-to-digital) conversion: The conversion of an analog signal into a digital equivalent. An A/D converter measures an input voltage and outputs a digitally encoded number corresponding to that voltage.

ASCII (American Standard Code for Information Interchange): A character set that includes the uppercase and lowercase English alphabet, numerals, special symbols and 32 control codes. Since each character is represented by a unique 7-bit binary number, one ASCII-encoded character can be stored in one byte of computer memory.

Asynchronous communication: Data transmission where the time interval between characters is allowed to vary.

Baud: Bits per second.

Baudot: A 5-bit character code. Since five bits only permit 2^5 or 32 permutations, this character set is limited in comparison to ASCII.

Buffer: Memory area in a computer or peripheral used for temporary storage of information that has just been received. The information is held in the buffer until the computer or device is ready to process it. Hence, a computer or device with memory designated as a buffer area can process one set of data while more sets are arriving.

Burning: The process of programming a read-only memory.

Character set: The repertoire of characters that an output device can display or print. Two common character sets are 96-character ASCII, which includes all ASCII characters except the 32 control codes, and 64-character ASCII, which includes all ASCII characters except the control codes, the lowercase alphabet and several other symbols.

Checksum: A method of detecting errors when information is being loaded into a computer from magnetic tape or paper tape. The checksum is the sum of the numerical values of the bytes on the tape. As the tape is loaded, the checksum is computed. After the loading is complete, this value is compared with the checksum value that was placed on the tape when it was generated. If the two are equal, the information is assumed to have been loaded without error.

Control character: Any of the 32 ASCII control codes. Their functions range from generating a carriage return to controlling remote devices.

cps (characters per second).

Crash: A system shutdown caused by a hardware or software malfunction.

CRC (cyclic redundancy check): An error-detection scheme (usually hardware implemented) that is often employed in disk devices. Although the mechanics are different, it is similar in principle to the checksum method. When information is stored, a CRC value is computed and stored. Whenever it is reread, the CRC value is computed once again. If the two values are equal, the information is assumed to be error-free.

Crowbar: A circuit that protects a system from dangerously high voltage surges.

CRT (cathode ray tube) terminal: A type of communications terminal that displays its output on a television-like screen. Synonym for video terminal.

Current loop: A type of serial communication where the presence or absence of an electrical signal (usually 60 mA) indicates the state of the bit being transmitted.

Cursor: A symbol on the display of a video terminal that indicates where the next character is to be located.

Device: A computer peripheral or an electronic component.

Diskette: A floppy disk.

Disk storage: A type of mass memory in which information is stored on a magnetically sensitive rotating disk. Disk drives are generally both faster and more expensive than paper tape or magnetic tape devices.

EBCDIC (Extended Binary Coded Decimal Interchange Code): An 8-bit character code used primarily in IBM equipment.

File: A group of related information records that are treated as a unit. The records may consist of data or program instructions.

Floppy disk: A type of disk storage that uses flexible disks made of a material similar to magnetic tape. Also refers to the disk itself.

FSK (frequency shift keying): A method of data transmission in which the state of the bit being transmitted is indicated by an audible tone. Three common types of FSK are those used by acoustic couplers (2225 Hz for logic 1, 2025 Hz for 0), radio teletypewriters (2125 Hz/2975 Hz) and the Kansas City cassette format (2400 Hz/1200 Hz).

Full duplex: Communication where data may be simultaneously transmitted and received by both ends of the circuit.

Graphics terminal: A video terminal capable of displaying user-programmed graphics.

Half duplex: Communication where only one end may transmit at a time.

Handshaking: The exchange of a sequence of signals required to complete an I/O operation.

Hard copy: Output printed on a permanent medium, such as paper.

Hard disk: Disk storage that uses rigid disks rather than flexible disks as the storage medium. Hard-disk devices can generally store more information and access it faster. Cost considerations, however, currently restrict their usage to medium- and large-scale applications.

Hard sectoring: Defining the sectors on a disk through hardware.

Impact printer: A printer that prints characters by mechanical means, such as a type ball (as opposed to thermal or ink jet methods).

Interface: A device that links two other devices. It converts signals from one into a format

that can be processed by the other.

Joystick: A type of input device. It has a stick that is manipulated by the user to produce different inputs. Joysticks are often used in conjunction with graphics terminals.

Kansas City Standard: A low-speed cassette storage format.

Light pen: An input device used in conjunction with a video display. When the user touches the display screen with the light pen, the electronics associated with the pen will determine the coordinates of the point that the user touched. These coordinates will then be transmitted to the computer.

Line printer: An output device that prints an entire line of information at a time.

lpm (lines per minute): Usually used to describe the speed of a line printer.

Mag tape: Magnetic tape, similar to that used by audio tape recorders, on which information can be stored in a computer-readable format.

Mark sense: A data-input method where the user designates information by placing pencil marks on cards. The cards are then fed into a special reader that translates the marks into a format that can be understood by the computer.

Mass storage: Auxiliary or bulk memory as opposed to main memory. Disk drives and tape drives are common mass-storage devices.

Modem (modulator-demodulator): A device used with frequency-shift-keying (FSK) data transmission. It converts FSK codes into their digital equivalents and vice versa.

Noise: An unwanted signal.

Noise immunity: A device's ability to accept valid signals while rejecting unwanted signals.

OCR (optical character recognition): Computer recognition of printed characters.

Off-line: A device that is not connected directly to its host computer. A keypunch is an example of an off-line device.

On-line: A device that is connected directly to its host computer.

Paper tape: A length of narrow paper punched in a pattern decodable by a paper-tape reader. Paper tape is bulkier than magnetic forms of data storage. On the other hand, paper tape cannot be erased accidentally and is easier to edit by splicing.

Parallel I/O: Data transmission where each bit has its own wire. All of the bits are transmitted simultaneously, as opposed to being sent one at a time (serially).

Parity check: An error-detection scheme in which an additional bit, the parity bit, is appended to each word or byte. Under even parity, the parity bit is 1 if there is an even number of 1s in the rest of the word. Under odd parity, the parity bit is 1 if there is an odd number of 1s in the rest of the word.

Peripheral: A unit, such as a communications terminal, that is external to the system processor.

Plotter: A hard-copy device that produces line drawings such as X/Y graphs. The coordinates of the points or lines to be plotted are normally supplied by the computer.

Port: A communication channel between a computer and another device.

Resolution: The density and overall quality of a video display. Also refers to the number of distinct points that can be plotted by a graphics terminal.

Response Time: The amount of time required for a computer to respond to an input from one of its terminals.

Roll-over: A property of some keyboards. Keys may be depressed in more rapid succession on a keyboard with roll-over.

RS-232: A standard for serial data communication.

RTTY (radio teletypewriter): Radio communication between teletypewriters.

Scrolling: A property of some video terminals. If the screen of such a video terminal is filled, it will move the entire display image upwards; the top line of text will be lost; and a blank line will appear at the bottom.

Serial I/O: Data transmission in which the bits are sent one by one over a single wire.

Soft copy: Output printed on a video display.

Soft sectoring: Defining the sector format of a disk through software.

Synchronous communication: Data transmission where the bits are transmitted at a fixed rate. The transmitter and receiver both use the same clock signals for synchronization.

Terminal: A device for communication with a computer. A typical terminal consists of a keyboard and a printer or video display.

Thermal printer: A hard-copy device that produces output on heat-sensitive paper.

TTY (teletypewriter): A hard-copy terminal.

TVT (television typewriter): A basic or low-end video terminal.

UART (universal asynchronous receiver-transmitter): An integrated circuit that converts parallel input into serial form, or vice-versa.

Video monitor: A device that is functionally identical to a television set, except that it has no channel selector. It receives its picture signal from an external source such as a video terminal board.

Voder: A speech synthesizer.

Software

Absolute address: The actual address of a memory location, as opposed to a relative address, which would not be known until the execution of the program.

Absolute assembler: An assembler that produces code in which all address references are absolute addresses.

Accumulator: A processor register used as intermediate storage for arithmetic operations.

Algorithm: The procedure used for performing a task.

Applications program: A program dedicated to a specific purpose or application. Applications programs are considered distinct from systems programs.

Argument: A value passed to a subroutine or function.

Assembler: A systems program that translates assembly-language programs into executable machine-level code.

Assembly language: A low-level programming aid that permits the programmer to use mnemonics instead of numerical op codes and alphanumeric labels instead of absolute memory addresses.

BCD (binary-coded decimal): A method of encoding decimal digits in the form of 4-bit binary numbers.

Bootstrap: A short loader program that loads a more sophisticated loader into memory. That loader, in turn, loads the desired program. The term bootstrap arises from the idea that the computer is picking itself up by its bootstraps. In other words, it progresses from the bootstrap to the loader to the main program itself.

Branch: See *jump*.

Breakpoint: A debugging aid. When a breakpoint is encountered in a program, execution of the program temporarily halts.

Bug: A programming error. Also refers to the cause of any hardware or software malfunction.

Call: An instruction that invokes a subroutine.

Chaining: The process of having one program transfer control to another program.

Code: Computer software.

Compiler: A systems program that translates high-level language programs into binary machine-level code.

Conditional jump: An instruction that executes a jump only if a certain condition is true.

Cross-assembler: An assembler that runs on a machine other than the one for which it is designed to assemble code.

Cross-compiler: A compiler that runs on a machine other than the one for which it is designed to compile code.

Counter: A register or memory location used to count the number of times a certain event occurs.

Database: A collection of information in a form that can be manipulated by a computer.

Debugging: The process of searching for and removing bugs.

Diagnostic: A systems program used as a hardware troubleshooting aid.

Disassembler: A systems program that converts machine-language code back into as-

sembly language.

DOS (disk operating system): A systems program that controls a disk system.

Driver: A program that controls (or drives) a device.

Dyadic: An operation that uses two operands.

Editor: A program that facilitates the editing of textual material or computer software.

Emulator: A program that allows one processor to simulate the instruction set of another processor.

Execute: To perform a computer instruction or run a program.

Executive: See *monitor.*

FIFO (first-in, first-out): A stack arrangement.

Fixed-point arithmetic: Arithmetic where the decimal point always remains at a predetermined position. Integer arithmetic is a type of fixed-point arithmetic, because the decimal point is always to the right of the mantissa.

Flag: A bit whose state signifies whether a certain condition has occurred.

Floating-point arithmetic: Arithmetic where the decimal point may occupy any position.

Flowchart: A diagram representing the logic of a computer program.

Hexadecimal: The base-16 number system.

High-level language: Computer language that allows the programmer to write programs using verbs, symbols and commands rather than machine code. Some common high-level languages are:

- ALGOL (ALGOrithmic Language)—Used primarily in scientific applications. Has gained much wider acceptance in Europe than in North America. Dialects include MAD and JOVIAL.
- APL (A Programming Language)—Emphasizes operations on groups of data. Most operations are designated by a single symbol rather than a textual verb.
- BASIC (Beginner's All-purpose Symbolic Instruction Code)—Oriented toward beginners rather than experienced programmers. Numerous incompatible versions exist.
- COBOL (COmmon Business-Oriented Language)—Used primarily in business applications.
- FORTRAN (FORmula TRANSlator)—The first high-level language. Emphasizes algebraic operations. Used primarily in scientific applications.
- PL/I (Programming Language/One)—Possesses qualities of many languages, especially ALGOL.
- PL/M (Programming Language/Meta)—A subset of PL/I. Usually implemented as a cross-compiler.
- SNOBOL—Emphasizes string operations.

Instruction set: The repertoire of inherent operations that a processor can execute.

Interpreter: A systems program that executes high-level language programs instruction by instruction. Interpretive language processors are considered distinct from compilers, which convert the entire program directly into machine-level code.

Interrupt: Temporary suspension of normal system operations while the processor responds to a request from another device.

Jump: An instruction that causes the processor to transfer control to another instruction.

Language processor: A systems program, typically either a compiler or an interpreter, that permits a computer to execute code written in a high-level language.

LIFO (last-in, first-out): A stack arrangement.

Loader: A program that loads information into main memory.

Loop: A program segment that is executed several times in a row.

Machine language: Binary code that can be directly executed by the processor, as opposed to assembly or high-level language.

Macro: See *pseudo-op.*

Macro assembler: An assembler that allows the programmer to define his own macros.

Matrix: A group of numbers organized on a rectangular grid and treated as a unit. The numbers can be referenced by their position on the grid.

Microcode: Software that de-

fines the instruction set of a microprogrammable processor.

Microprogrammable processor: A processor whose actual instruction set is not accessed by the programmer. Instead, another, more versatile, instruction set is simulated by microcode.

Mnemonic: An abbreviation for a computer instruction. For example, jump to subroutine might be represented by the mnemonic JSR.

Modulo: An arithmetic function. A number x evaluated modulo n gives the integer remainder of x/n. For example, 100 modulo 97 equals the remainder of 100/97, or 3.

Monadic: An operation that uses only one operand.

Monitor: A systems program that controls I/O and related functions.

Multiprocessing system: A system that includes more than one processor, each of which works concurrently on separate tasks.

Multiprogramming system: A system that can execute more than one program at a time.

Multitasking system: See *multiprogramming system*.

Nesting: The practice of placing one loop inside of another or having one subroutine reference another.

Niladic: An operation for which no operands are specified.

Nybble: A group of four bits, or one-half of a byte.

Object code: Machine-language code.

Octal: The base-8 number system.

Op code (operation code): A number or mnemonic that specifies the instruction to be executed.

Operand: The value being processed by the operator. For example, in the expression 2+3 the operands are two and three.

Operator: A symbol or command that specifies the operation to be performed.

OS (operating system): See *monitor*.

Overlaying: A technique used to increase the apparent size of main memory. This is accomplished by keeping only the code or data that is currently being accessed in main memory. The rest is kept on a mass-storage device until needed.

Page: A segment of memory that is treated as a unit. For example, 65,536 bytes of memory (64K) might be divided into 16 pages of 4096 bytes.

Parsing: Analyzing a character string and breaking it down into a group of more easily processed components.

Pass: A scanning of source code by an assembler or a compiler. For example, a two-pass assembler is one that processes the source code in two separate steps.

Polling: The interrogation of the devices in a system by the processor to determine if and where any I/O operations are pending.

Program: A sequence of instructions that permit a computer to perform a task.

Pseudo-op (pseudo-operation): An instruction that is implemented by an assembler but is not in the processor's instruction set. It is often used to give information to the assembler. Or, if the pseudo-op is part of the program itself, the assembler will assemble in its place a sequence of instructions that simulates the pseudo-op.

Queue: A waiting line of tasks or data.

Real time: A system function that is controlled by external events. For example, a system that reacts to inputs from a temperature sensor would be considered a real-time system.

Recursive subroutine: A subroutine that invokes itself. Or, a subroutine that invokes another subroutine, which, in turn, invokes the original subroutine again.

Reentrant subroutine: In a multiprogramming system, a subroutine of which only one copy resides in main memory. This copy is shared by several programs.

Register: A unit of memory that is contained within the processor circuitry itself, as opposed to main memory or mass storage.

Relative addressing: An addressing mode in which the target memory location is specified as the sum of a variable and a constant. The variable is referred to as the origin and is usually the address of the instruction currently being executed. The constant is specified within the instruction. For example, the relative address +10 is the same as the absolute address x+10, where x is the origin.

Relocatable code: Code that can occupy any position in main memory.

RPN (reverse Polish notation): A mathematical notation in which the operator is placed after its operands, rather than between them. For example, the expression 2+3 would be written 2,3+ in RPN.

Seed: An argument that is used as the initial value for a pseudo-random number generator.

Simulation: A computer program that mathematically models a process.

Single stepping: Having the computer execute a program slowly so that the user can watch each step and its effects.

Software: Computer programs.

Software license: A contract signed by the purchaser of a software product in which he is usually made to agree not to make copies of the software for resale.

Source code: Code that is to be processed by a systems program such as a compiler or an assembler.

Stack: A sequential data list stored in main memory. Rather than addressing the stack elements by their memory location, the processor retrieves information from the stack by popping elements from the top (LIFO) or from the bottom (FIFO).

Statement: A single computer instruction within a computer program.

String: A variable that contains alphanumeric text.

Subroutine: A program segment that may be invoked by another section of the program. A subroutine call differs from a simple jump instruction in that after completion of the subroutine, execution of the program will return to the program section that invoked it.

Systems program: A program that does not perform actual problem solving but rather is used to control system operations or act as a programming aid.

Time-sharing: See *multiprogramming.*

Trap: A hardware- or software-implemented function that signals the processor whenever a specified condition occurs.

Unary: See *monadic.*

USR (user service routine): A machine-language subroutine that may be invoked by a high-level language program.

Utility: A frequently used program or subroutine. Utility routines are most often associated with systems programs rather than applications programs.

Virtual device: An imaginary device that the processor assumes to be present. Automatic memory overlaying, or virtual memory, is an example.

Workspace: A loosely defined term that usually refers to the amount of main memory available for programs and data. Also, the APL name for all of the memory-resident functions and variables that the programmer has defined.

Miscellaneous

Alphanumeric: Alphabetic and numeric characters.

ANSI (American National Standards Institute).

Benchmark: A test that compares the performance characteristics of several devices, programs or systems.

Binary: The base-two number system. It uses only the digits 0 and 1. The numbers $0\text{-}4_{10}$ are written, 0, 1, 10, 11 and 100.

Bipolar: A type of circuit that uses conventional PNP or NPN transistors rather than FETs.

Breadboard: An electronics design aid that allows a circuit to be constructed, tested and modified more easily than with printed circuit or wire-wrapping techniques.

Bypass capacitor: A capacitor used to reduce electrical noise from the power supply.

CAI (computer-assisted instruction).

Chip: An integrated circuit.

CMOS (complementary MOS): A type of integrated circuit whose output structure consists of an N-type MOSFET and a P-type MOSFET in series.

Dedicated device: A device that is used exclusively for one function.
DIP (dual in-line package): A type of integrated circuit packaging. It is characterized by a rectangular shape and by pins that point downward. The pins are arranged on two sides of the rectangle.
Discrete component: An electronic component that contains only one function, as opposed to an integrated circuit.
FET (field-effect transistor): A unipolar transistor. That is, it contains only P-type or N-type doping (but not both).
Flat pack: A type of integrated-circuit packaging in which the pins extend outward, rather than pointing down as on a DIP.
Flip-flop: A circuit that changes its logical state when signaled to do so by another device.
Gate: A circuit that performs a Boolean logic operation.
Hacker: A computer enthusiast.
IC (integrated circuit).
Intelligent device: A device that contains its own processor.
I/O (input/output).
k: A unit of 1000.
K: A unit of 1024.
Kludge: Makeshift.
LED (light-emitting diode): A type of digital output display that is frequently used in calculators.
Linear IC: An analog integrated circuit, as opposed to a digital integrated circuit.
LSI (large-scale integration): The class of integrated circuits that contain the largest number of functions per chip. Microprocessors are LSI devices.
M (mega): A unit of one million (10^6) or 1024^2 (1,048,576).
Milli-: One thousandth (10^{-3}).
MOS (metal-oxide semiconductor): A semiconductor structure that is used in many FETs and integrated circuits, which are then referred to as MOSFETs and MOS ICs.
MSI (medium-scale integration): The class of integrated circuits having a density between those of LSI and SSI devices.
MTBF (mean time between failures): The length of time for which a device can reasonably be expected to operate without malfunction.
MTTR (mean time to repair): The length of time typically required to service a device following a breakdown.
Multilayer: A type of printed circuit board that has several circuit layers connected by electroplated holes.
NMOS (N-type MOS).
OEM (original equipment manufacturer): A manufacturer who buys equipment from other suppliers and integrates it into a single system for resale.
PC (printed circuit) board: A circuit board whose electrical connections are made through conductive material that is contained on the board itself, rather than with individual wires.
PLA (programmable logic array): A device (usually an integrated circuit) containing a set of logic gates whose interconnections may be programmed.
PMOS (P-type MOS).
Populated board: A circuit board that contains all of its electronic components. The components for an unpopulated board, conversely, must be supplied by the purchaser.
Power fail/restart: A facility that enables a computer to return to normal operation after a power failure.
Run time: The time at which the program is executed. Also, the amount of time required to execute the program.
Second source: A manufacturer who produces a product that is interchangeable with the product of another manufacturer.
SSI (small-scale integration): The class of integrated circuits that have the fewest number of functions per chip.
TTL (transistor-transistor logic): A family of integrated circuits characterized by relatively high speed and low power consumption. These are usually SSI devices, so named because of the dual-transistor arrangement used by the output stage.
Transparent: A process that is not visible to the user or to other devices. Transparent memory refresh is an example.
Vendor: Supplier.
Wire-wrap: A type of circuit board construction. Electrical connections are made through wires connected to the posts that correspond to the proper component lead. ■

✓ Reader Service—see page 179

An Introduction to Microfilming

Microfilming is a storage medium that you may have overlooked in your search for hard-copy devices. For the reasons outlined in this article, it may be worth your consideration.

Photo 1. Camera in place.

Michael Schwartz
1260 Doblon St., Apt. B
Green Bay WI 54302

I want to introduce you to one hard-copy device that may not cost you a cent: your camera. It is easy to point it toward your CRT terminal and start microfilming both programs and data.

Hobby Microfilming

Why hobby microfilming? The first reason is low cost. Most cameras can be used to give a readable image on film. You can even buy a nice single-lens reflex camera and enough film to keep you happy for years for less than the cost of the least expensive printer.

Second, document volume will be reduced. I can mail a large program in an ordinary envelope without even an extra stamp. By sticking a few slides of my current programming work along with a small folding slide viewer in my pocket, I can work inconspicuously even in

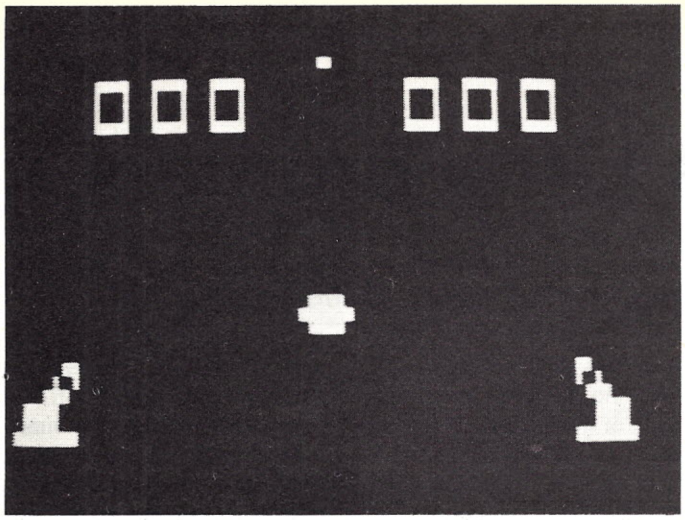


Photo 2a. Shot of a 9 inch TV screen.

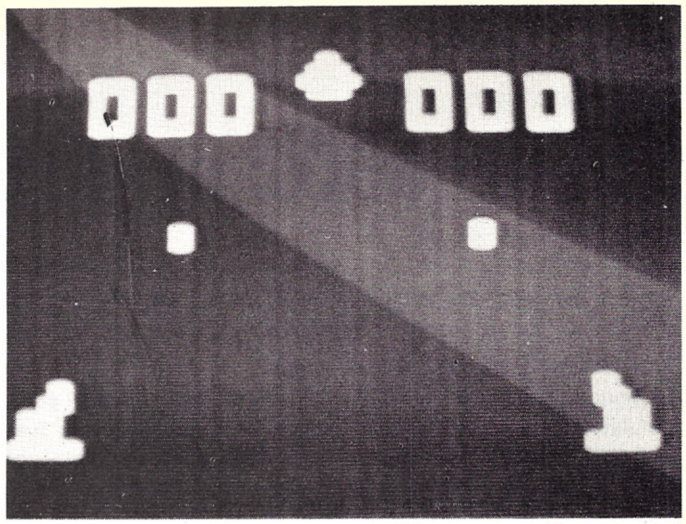


Photo 2b. Similar shot with contrast and brightness on full.

the tight quarters of a plane.

Third, black and white film has a more archival permanence than paper. Film does not fade, become musty or dog-eared and is not attacked by moisture as easily as paper.

What Camera to Use?

Although a single-lens reflex is the ideal camera, the basic problems encountered with any camera are those of exposure and focus. The exposure is not critical. Remember that we are not dealing with a normal photographic subject of delicate gray tones, but a pure black and white image. If you have an adjustable camera, set the shutter speed at 1/15 or 1/30 of a second and F/2.8 for ASA 125 film or F/4 for ASA 400. Experiment with different settings, but almost any setting in this area should produce a readable and, therefore, good image. As far as focus, most modern cameras focus down to two feet or less, so a close-up lens is not necessary on an adjustable camera.

Those of you with box cameras are not left out. Many models have close-up settings as well as "cloudy" or "shade" settings or even an electric eye. Your camera dealer can provide more information on close-up lenses or other attachments for your particular camera. Try what you have on hand, though, before investing in any new equipment.

Using the Camera

Set up the camera on a good sturdy support directly in front of the CRT screen (Photo 1). I tilt my screen to one side so I can mount my camera on a clamp attached to the side of the table. Adjust the brightness and contrast of your CRT so that the background is black but just on the verge of turning gray and the letters are white but not so bright that the image begins to flare out. Photo 2a shows a good setup using the worst case of an ordinary 9 inch TV set and an rf converter.

Photo 2b shows the same scene with brightness and contrast turned up all the way, the fluorescent room lights glaring on the top of the screen. Just for good measure, I scratched the wet negative with my fingernail. Photographic negatives will have reversed tones. If you have white letters on a black background on your CRT, the negatives will show black letters on a white background.

Sometimes I have a few extra exposures left on a roll of normal photographs and I use them up microfilming even if they are color slides. But when I have a lot of computer images to photograph I buy any bulk outdated black and white film. By developing it myself my cost per CRT image is well under one cent. If you send your film to a photo lab, be sure to specify "develop negatives only, no prints" on the envelope or you will have to pay for a lot of expensive prints that you don't want.

When the negatives come back, you can read them with a magnifying glass or mount them like slides (Photo 3). Slip-in (not heat sealed) slide mounts are easy to use and reuse. A jeweler's loupe (6x or 8x) or a pocket slide viewer is handy for portable use, but a tabletop viewer or slide projector will leave your hands free for more stationary locations.

I mount my microfilm in slide mounts unless it is just for a file copy. Your photo dealer can show you other methods of negative filing as well as a full line of slide viewing devices.

Photo Finish

I am mostly a software type, but I have been considering a system that could put a sequence of bits on the screen to be photographed and read into the system later, possibly via a bar scanning device or CCD chip. Many rapid developing systems are available to make this an on-line alternative with access times someplace between disks and mag tape.

In the business world, microfilm is growing by leaps and bounds because of high printing speed, paper saving and the other reasons I listed at the beginning of this article. It is high time the micro world got in on some of these benefits. ■

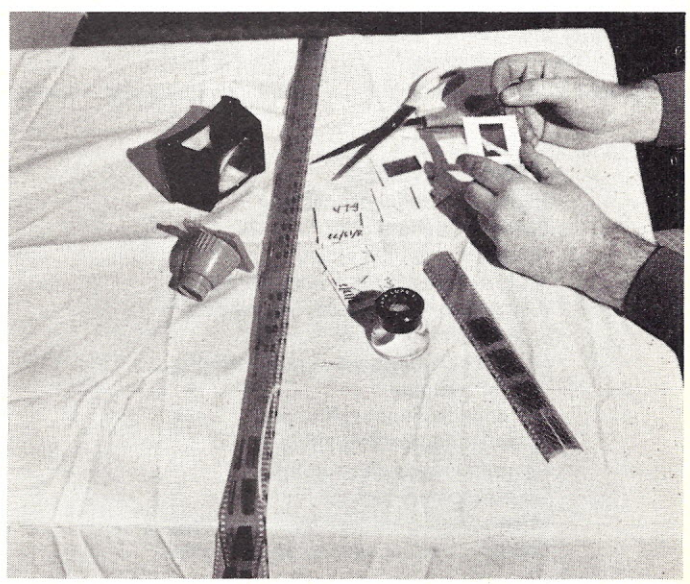


Photo 3. Preparing negatives.

Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

The 6502 and Its Little Brothers

If you haven't met them yet, here are the other members of the 6502 family.

The MCS6502 by MOS Technology is a well-known, popular microprocessor. That it is found in the PET, PAIA, KIM-1, Apple and OSI personal computers guarantees that much will be written about it and much software will be written for it. However, less well known are the other members of the MCS65XX family. Let's see how they compare and what they have to offer.

This article is an overview and not a tutorial (consult the references at the end of the article). The hardware manual, the software manual and the KIM-1 user manual, which retail for $30, are included with the purchase of the KIM-1.

General Features

While all of the members of the family share certain common characteristics, each has its own individual personality. Table 1 is a list of the features possessed by all of them. It has been said that the 6502 is closely related to the MC6800. In fact, it has been claimed by some writers that the 6502 is the second-generation 6800. Whether this is true or not, I'll leave up to the experts to argue.

Look at Table 1 while I elaborate on some of these features. Eight-bit parallel processing and the bidirectional bus are standard in the micro field, and the trend to +5 V dc-only requirements goes a long way toward simplifying power supply design and construction. The READY input allows the use of memory, which is slower than the processor.

Depending on the type of 65XX that is being used, an addressing capability of from 4K to 65K is available, as we shall see. Variations as to processor speed and clock types are also related to 65XX type. The architecture and instruction set allow decimal or binary arithmetic, 13 addressing modes, a programmable stack pointer, direct memory access and maskable or non-maskable interrupts. An output clock is provided for synchronizing the operation of support chips. All nine types recognize the same instruction set, which means that they are completely software compatible.

Specific Types

Table 2 shows the various members of the 65XX family and the features that make each of them unique. Each version is available for operation with either a 1 or 2 MHz clock. The 2 MHz types are suffixed with an "A" (6502A, 6512A, etc.) for identification purposes.

The 6502, 6503, 6504, 6505 and 6506, which operate from an internal clock, are aimed at high-performance/low-cost applications. The 6512, 6513, 6514 and 6515 require an external clock and are intended for multi-processor applications where all of the CPUs must march to the beat of the same drummer. The major differences between the family members relate to their memory-addressing ability and interrupt capabilities.

The 6502 and 6512 come in 40-pin DIPs (dual in-line package), while the rest are packaged in a 28-pin DIP, which does not provide enough pins to support all 16 address lines. This restriction is obviously deliberate, and these limited versions are intended for dedicated controller applications where access to great amounts of memory is not usually needed. What memory is required is usually in ROM or PROM.

Architecture

Fig. 1 is a block diagram of the 65XX internal makeup. The accumulator (A) handles eight bits of data at a time. The eight-bit X and Y registers are used in the indexed addressing modes and can also be used as storage and counting registers during the execution of a program.

The 16-bit Program Counter (PC) can access 65K bytes of memory (with the restrictions on some versions due to the lack of all 16 address leads). The 8-bit programmable Stack Pointer (S) always points to page one (0100 to 01FF hex).

The Processor Status (P) register contains flags that signal the occurrence of special events and can be used to direct conditional branching or

- MC6800 compatible
- Eight-bit parallel processing
- Bidirectional data bus
- Requires +5 V dc only
- Will wait for slow memory (READY)
- Addresses for 4K to 65K of memory
- 1 or 2 MHz operation
- Choice of internal or external clocks
- 56 instructions
- Decimal or binary arithmetic
- Thirteen addressing modes
- Programmable stack pointer
- Direct memory access
- Maskable and non-maskable interrupts
- Output clock for timing support chips

Table 1. List of the features that are common to all members of the MCS65XX microprocessor family.

Type	Package	Clock	Features
6502	40-pin DIP	Internal	All
6512	40-pin DIP	External	All
6503	28-pin DIP	Internal	Can address 4K—maskable and non-maskable interrupts
6513	28-pin DIP	External	Can address 4K—maskable and non-maskable interrupts
6504	28-pin DIP	Internal	Can address 8K—maskable interrupt
6514	28-pin DIP	External	Can address 8K—maskable interrupt
6505	28-pin DIP	Internal	Can address 4K—maskable interrupt—READY lead
6515	28-pin DIP	External	Can address 4K—maskable interrupt—READY lead
6506	28-pin DIP	Internal	Can address 4K—maskable interrupt—2 phase clock output

Table 2. Features of the individual members of the 65XX microprocessor family. Each would fill certain requirements in a general or dedicated controller application.

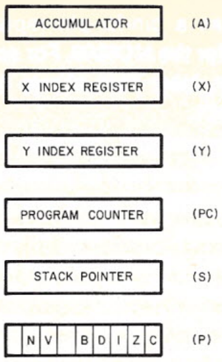


Fig. 1. The six registers contained within the MCS65XX microprocessors. Each is eight bits wide, except the Program Counter (PC), which is a 16-bit register.

indicate processor status. For instance, the Negative (N) flag is set to a 1 whenever the result of an operation is negative. The Branch on Minus (BMI) and Branch on Plus (BPL) use this flag to determine whether or not to branch.

The Zero (Z) flag is set if the result of an operation is zero, otherwise it is reset (0). The Branch on Zero (BEQ) and Branch on not Zero (BNE) instructions pivot on this flag.

The Branch on Overflow Set (BVS) and Branch on Overflow Clear (BVC) use the Overflow (V) flag, and the Branch on Carry Set (BCS) and Branch on Carry Clear (BCC) instructions utilize the Carry (C) flag in a like manner.

The Interrupt Disable (I) flag is set by the programmer (SEI) or the processor when interrupts are undesirable, such as during resets and previous interrupts. The Decimal Mode (D) flag is set by the programmer (SED) when the BCD (binary coded decimal) form of addition or subtraction is desired rather than the binary form.

And finally, the Break Command (B) flag is set by the processor whenever a Break instruction (BRK) has been executed. There is an eighth bit in the Processor Status register that is not being used.

Instruction Set

The 65XX instruction set consists of 56 different types, and since quite a few of these types can take more than one form,

ADC	Add memory to Accumulator with carry
AND	"AND" memory with Accumulator
ASL	Shift left one bit
BCC	Branch on Carry clear
BCS	Branch on Carry set
BEQ	Branch on result equal
BIT	Test bits in memory with Accumulator
BMI	Branch on result minus
BNE	Branch on result not zero
BPL	Branch on result plus
BRK	Force break
BVC	Branch on Overflow clear
BVS	Branch on Overflow set
CLC	Clear Carry flag
CLD	Clear decimal mode
CLI	Clear Interrupt disable
CLV	Clear Overflow flag
CMP	Compare memory and Accumulator
CPX	Compare memory and Index X
CPY	Compare memory and Index Y
DEC	Decrement memory
DEX	Decrement Index X
DEY	Decrement Index Y
EOR	"Exclusive OR" memory with Accumulator
INC	Increment memory
INX	Increment Index X
INY	Increment Index Y
JMP	Jump to new location
JSR	Jump to subroutine
LDA	Load Accumulator with memory
LDX	Load Index X with memory
LDY	Load Index Y with memory
LSR	Shift one bit right
NOP	No operation
ORA	"OR" memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Process Status on stack
PLA	Pull Accumulator from Stack
PLP	Pull Process Status from stack
ROL	Rotate one bit left
ROR	Rotate one bit right
RTI	Return from interrupt
RTS	Return from subroutine
SBC	Subtract memory from Accumulator with borrow
SEC	Set Carry flag
SED	Set decimal mode
SEI	Set interrupt disable
STA	Store Accumulator in memory
STX	Store Index X in memory
STY	Store index Y in memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index to Accumulator
TXS	Transfer Index X to Stack Pointer
TYA	Transfer Index Y to Accumulator

Table 3. Mnemonics and descriptions for the 65XX instruction set.

there are 146 op codes available to the programmer. Most of these variations have to do with addressing modes. Table 3 is a list of the instruction types, and Table 4 spells out the addressing modes.

This multiplicity of available addressing modes can make the programmer's life easier or harder depending on your point of view. Certainly the zero page, indexed and relative modes make for simpler, shorter code and make it easier to relocate programs due to the lack of a need for absolute addressing in most cases.

The 65XX series are memory-oriented processors as opposed to the 8080 and Z-80, which are register oriented.

Accumulator
Immediate
Absolute
Zero Page
Indexed Zero Page
Indexed Absolute
Implied
Relative
Indexed Indirect
Indirect Indexed
Absolute Indirect

Table 4. Addressing modes available to the 65XX programmer. Since these are complex, I suggest the MOS Technology Programming Manual for a thorough explanation.

This means that most instruction results must be stored in memory, as in the 6800. The 8080 and Z-80 store most results in internal registers. Also, like the 6800, the 65XX has no specific provision for I/O addressing. I/O locations are handled exactly the same as memory locations. Therefore, any instruction that relates to memory operations will do the same for I/O.

Applications

The 65XX CPU family was designed with many different applications in mind. As I mentioned earlier, many general-purpose personal computers use the 6502. However, each member of the family is seeking its own little niche.

One computer that uses the 6503 is the PAIA 8700. The 8700 is designed to interface with a music synthesizer, but many control applications are certainly possible. More and more of the latest peripherals, such as floppy-disk interfaces, high-speed printers, etc., contain a dedicated controller . . . not to mention the kitchen ranges, washers, dryers, clocks and cars that have them now and the many more home appliances that will in the near future.

How about a subordinate processor with limited memory that acts as a printer buffer. Instead of your main CPU dumping one character at a time to the printer using the printer's time frame (slow!), why not dump a 256- or 512-byte block on the subordinate CPU to un-

MCS6520

Compatibility with MCS65XX microprocessors
Two 8-bit bidirectional ports
Two programmable Data Direction Registers
Four individually controlled Interrupt input lines
Two peripheral control lines
Handshake logic for input and output peripheral control
Tri-state input/output lines
8-bit bidirectional Data bus
Two programmable Control Registers
Program controlled interrupt capability

MCS6530

Compatibility with MCS65XX microprocessors
Two 8-bit bidirectional ports
Two programmable Data Direction Registers
1024 × 8 bit ROM
64 × 8 bit static RAM
Programmable Interval Timer
Programmable Interval Timer interrupt
TTL- and CMOS-compatible input/output lines
Tri-state input/output lines
8-bit bidirectional Data bus

Table 5. Features of the MCS6520 peripheral interface device and the MCS6530 peripheral interface-memory-timer device.

load on the printer. When that has been printed, an interrupt can signal the need for a new block. Meanwhile, the main CPU is handling other tasks. CPU prices have come down to the point where this is a feasible consideration.

Maybe you would consider an auxiliary CPU-driven real-time clock that is ready with the time whenever your programs need it without the need for the main CPU to be continually interrupted to update that clock.

You receive a bonus if your main and subordinate CPUs belong to the same family. You can use the assemblers, text editors, etc., on your big system to write programs for little brother. This is helpful since most of the controller-type applications don't allow elaborate keyboards, video terminals, etc. So you write your subordinate control software using the big gun, load it into RAM or burn it into PROM and free the big machine for the complex tasks.

A computer music system in which the music is composed on the big machine and played by the little one is another possibility. The main machine would contain the software that generated the code that the subordinate CPU executed.

Peripherals

There are two other members of this family that we haven't mentioned yet—the MCS6520 and 6530. The MCS6520 is a peripheral interface device and the MCS6530 is a peripheral interface-memory-timer device. Table 5 shows their features.

The 6520, which is pin-for-pin compatible with the MC6820 PIA (peripheral interface adapter), contains two eight-bit bidirectional parallel ports and four interrupt lines. Two of these lines can also be used as peripheral control lines.

The 6530 also contains two eight-bit bidirectional parallel ports but has other features as well. These include 1024 × 8 bit ROM, 64 × 8 bit RAM and a programmable interval timer that has the ability to interrupt the CPU or not as desired.

Two 6530s are utilized in the KIM-1. They provide the ROM operating system, RAM scratchpad, two timers and four ports. Two of the ports, one timer and part of the RAM are used by the operating system, leaving the rest for the programmer's use.

Conclusion

Early MOS Technology literature (1975) describes the MCS 6501 as a pin-for-pin replacement for the MC6800. For some reason, later data sheets omit the 6501.

Programmers familiar with the 6800 will feel at home with the MCS65XX, while the 8080 aficionado will face a whole new way of doing things. Most noticeable to the latter will be the lack of numerous registers that he used to use for storing, pointing and counting. He will also notice all of the extra addressing modes that the 65XX series possesses. The Z-80 contains some of both worlds.

MOS Technology has attempted to take the basic architecture of the MCS6502 and repackage it to fit various requirements. Have they been successful in this attempt? Only the end-user can answer that. ■

References

MOS Technology, 950 Rittenhouse Rd., Norristown PA 19401: *MCS6500 Programming Manual, MCS6500 Hardware Manual, KIM-1 User Manual, MCS6500 MPU Data Sheet, MCS6520 and MCS6530 Data Sheets* and *MCS6500 Instruction Set Summary Card.*

Addison-Wesley Publishing Co., Reading MA 08167: *Programming a Microcomputer: 6502*, Caxton Foster.

PAIA Electronics, 1020 W. Wilshire Blvd., Oklahoma City OK 73116: *PAIA 8700 Computer/Controller Assembly and Using Manual.*

✓ Reader Service—see page 179

Kilobaud Classified

Kilobaud Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment or software. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 37 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or not applicable advertisements. Mail copy with payment to: **Kilobaud Classified, Kilobaud, Peterborough NH 03458.** Do not include any other material with your ad as it may be delayed.

Smoke Signal Broadcasting & SWTP owners: Software now available for Chieftain I & II microcomputer systems & SWTP 5" disk systems. Random accessing software: A/R Inven, $370; A/P, $270; P/R, $290; G/L, $290; F/A Inven system, $200. Application manuals are $20 each if purchased separately. Steve Chavez, 2920 Carlisle N.E., Albuquerque NM 87110.

H8 Owners: Save and restore disk files to and from cassette. Program tape and instructions, $15. Double media disks, $7.50. Karl K. Sidinger, 6967 W. Rancho Dr., Glendale AZ 85303.

TRS-80 Software: Business programs written in 16K Level II. Available for cassette or disk files. Send for list & prices. L Owens, Rt. 6, Box 336A, Thomasville GA 31792.

TRS-80 Yellow Page. A free catalog of software & hardware. For fast delivery, TRS-80 owners send 2 SASE envelopes and $1. Vendors send $20 & full prod. descrip. to share printing costs. Features business software & word processing. Write to: 96 Dothan St., Arlington MA 02174.

KISS is money making project via comp dating for schools, clubs. TRS-80 or any string BASIC 16K holds 400 surveys. (net $100+) List, doc., surveys, admin. instr. for $7. D. Bohlke, Coggon IA 52218.

Apple II Checkers, Tictactoe on cassette. Board shows all moves. Both for $10. Others. Send SASE for free list. Ed Dixon, Progress St., Florence SC 29501.

---

*Books of computer games in BASIC. ENIGMAS-1 ($8): Gone Fishing, Concentration, Starship, Craps, Slot-Machine, Sherlock Holmes, Tank Attack. ENIGMAS-2 ($8): Number Guess, Mortar Battle, In-Between, Shell Game, Safari, Starship-2, Dice Roll, Puzzle. *Catalog and test program Frog Race, $1. Available in Standard BASIC, SWTP 8K BASIC or Radio Shack TRS-80 BASIC. Please specify. B. Erikson, PO Box 11099, Chicago IL 60611.

TRS-80 Owners! Tired of endlessly thumbing through your Level II manual? Send $1 and a self-addressed, stamped envelope and receive a complete (over 275 entries!!) Level II Manual Index. You will not believe how much time this will save you. Send $1 and SASE to Dave Rose, 3439 Harding Ave., Hapeville GA 30354.

Original PET Software! A variety of action games with graphics and sound effects, useful math routines, interfaces (Teletype, parallel printers and telephone) and a program for creating, saving and playing back music. Send for a free catalogue today. Peter Ruetz, 368 Albion Ave., Woodside CA 94062.

PET: "Mill" a complex strategic board game from Europe, uses 8K. Board, men, moves, etc. on screen w/full graphics. Info, & game programs on cassette, $8. R. Prestien, 6278 SW 14 St., Miami FL 33144.

TRS-80 Sticky Relay Problems. Instructions for the correct fix. For all TRS-80s, old and new. This one works. Send $2 to Jim Creasy, PO Box 722, Pottstown PA 19464.

Cassettes: quality G.E., leaderless. 10 min., shipd prepaid UPS. 10 for $6; 50 for $27.50; 100 for $50. Rudolph Nelson W1HWY, POB 421, Merrimack NH 03054.

TRS-80 Owners—Small business software. Ledger posting program, gives totals by month, yrtd., allows printout for verification or account statements. Inventory control pgrm., flexible for your needs. Check posting pgrm., gives totals by category, month & yrtd. Plus more. Send SASE for more details & prices. Level II 16K tape or disk. R. Lynne, PO Box 447, Carrollton TX 75006.

TRS-80 Telephone Dialer. Level II 16K program enables the computer to dial numbers from a directory in memory. Single-key redialing. Requires simple hardware installation. Cassette and complete instructions, $12.95. J. H. Nestor, 39114 Route 303, Grafton OH 44044.

PLAY $$$ONOPOLY. World famous board game for TRS-80 16K Level II. 1 to 4 players. Full graphics, $12.95 U.F. Racine, 445 S. 72, Kansas City KS 66111.

For Sale: Heath H8 w/48K, 2 ea SIO/cass. int.; $1975. Heath WH17 dual floppy-disk system; $925. Heath H9 video terminal; $550. Heath cass. pl/rec; $45. All factory tested and running—some under fac. warranty. Reason selling: I have 2 computer systems. Buy package for $3300 or separately. All offers considered. Ray King, 915 El Rancho, Pocatello ID 83201. (208) 237-0979.

Digital Group Z-80 system. 32K in beautifully matched cabinets by DG. Parallel IO, built to handle 64K memory. Includes CRT, keyboard, 4 PhiDeck cass. recorders and *note*: an unused DecWriter II Printer. Up & running except printer. All software including Business BASIC & documentation. Ideal for all applications. Price $5000. DeGraff Associates, PO Box 18640, Salt Lake City UT 84118. Phone (801) 966-1451.

TRS-80 SOFTWARE DIRECTORY Over 800 programs are listed in a directory of TRS-80 software. Programs are listed by company and classification. The company name and address are given as well as the price. Over 125 companies are listed. Send $4 to: H. DiBlasi, Box 1664, Lake Havasu AZ 86403.

Quality TRS-80 Programs. Business, games, ham radio, Monopoly, Yahtzee, many more. Send SASE for information and complete list. Sandy Stange, Box 145, Lithonia GA 30058.

Adventure for TRS-80. $14.95 per Adventure; three Adventures available or send SASE for info. M. Scott Adams, PO Box 3435, Longwood FL 32750.

Have a printer? Make big money in shopping malls, schools, etc. selling biorhythm charts! Send $2 for my TRS-80 Lev II, 4K program listing for generating biorhythm charts. Sample chart and documentation incl. N. W. Lewis, Box 8384, Metairie LA 70011.

OSI Challenger III—dual floppies, 32K RAM, 3 processors. FDOS, BASIC, assembler, disassembler, word processing. Original cost $3700, sell for $3100. Bill Gale, 439 S. Orange Ave., S. Orange NJ. (201) 762-8279.

I make flat cable assemblies to interface your Radio Shack, Altair, Heath, DEC, etc. Spec. type & length. Call anytime. (203) 281-1331.

Bally Owners who subscribe to Arcadian know how to get four colors on the screen, do floating-point arithmetic and really get into making Tiny BASIC jump by poking it into the right places. Annual subscr., incl. 5 back issues, is $5 to Robert Fabris, 3626 Morrie Dr., San Jose CA 95127.

Audio tape interface, 250 baud, compatible w/L-I TRS-80, 256 bytes Z-80 machine code, list & hardward diagr., $3. Richard c/o ENVO, 800 Follin Ln., Vienna VA 22180. (703) 281-5515.

Morse code on TRS-80. Practice/send. Cassette or disk, $20. WB0QNT, Box 86, Douglass KS 67039.

For Sale: Minicomputer & Software. 64K CPU, 10M byte disk, 5 fix, 5 removable, 165 cps printer, 2 CRTs, 10 disk packs, up to 8 CRTs, 300 lpm printer, more disk & core too. Software—over 100 programs for warehouse distribution, order entry, invoicing, A/R, inventory control, sales analysis, history & commissions, systems diagnostics, many games. $30K. David Mayer, PO Box 1494, Englewood Clfs NJ 07632.

TRS-80 Users in Europe. We sell a great number of games, educational and scientific software and a "TRS-80 Resource Book" with programming tips, hardware/software information. Send stamped envelope for more details. Vortex 80, Postfach 401606, 8 Munchen, West Germany.

TRS-80 Screen Printer Cat. 26-1151. Functions with or without interface. Cost $599. Sell $299, mint condition. Cable & roll included. (914) 783-3930.

TRS-80 Level II, 16K. For the finest 16K Space Treck available (includes 4K 'Mission Briefing' plus 16K 'Combat Program'), send $14.95 to: Bill Vitulli, 4309 White Plains Rd., Bronx, NY 10466.

Pennywhistle to Nickelwhistle Conversion. Convert your Pennywhistle modem to full duplex Originate and Answer modes. Uses only a few passive parts. Plans—Only $10. Ted Wolff, 579 W. 215th St., NYC NY 10034.

KIM-1 Single Pass Editor Assembler. No add-on hardware or memory req'd. 150 lines off tape or 50 resident on std. KIM. Add min. 340 & 120, respectively, for each K of addit'l memory. MOS Tech mnemonics, compatible labels/var. 18 pg documentation delivered immed. Super fast, flexible, efficient, economical. $6.50 check/MO to: T. Davis, 1521 Riverside Dr., Rainelle WV 25962.

Tiny-C & Assembler programming bulletin for TRS-80. $1 brings you issue plus BASIC, Asm and System tape formats. R. Varty, 2193 Haygate Cr., Mississauga, Ont., Canada L5K 1L7.

Teletype Users: Unique solid state time delay relay. Reduces energy and maintenance costs. Info 50¢; Plans $5; with PC $10. Keith Ryan, Box 3103-K, Ottawa, Ontario, Canada K1P 6H7.

PET for sale: $650 cert. check—8K, e/w expndr mite printer interface, PET & mite printer $1k, E Comer, 4578 Hearthstone Ct., St. Mtn. GA 30083. (404) 296-1403.

TRS-80 Level II 16K. Educational games with exciting graphics. Delight all kids (7 to 70). Kids and adults learn programming by making suggested changes. Thirty programs on cassette; $5, postpaid. Home and classroom tested by Ph.D. Free 32-page catalog has English, biology, chemistry and math programs. Set of classroom guides $1. Indiana Biolab, Palmyra IN 47164.

TRS-80 Hardware Genius? TRS-80 Software Expert? Are you one but not both? With my schematic, parts list and software listing you can build an excellent RS-232 printer interface which sells for up to $75 for only a few dollars and some spare time. Send only $5 for schematic, parts list and software listing to Bruce Taylor, 118 South Mill St., Pryor OK 74361.

OEA OP-80A paper tape reader: Reads fast as you can pull tape thru. Assembled with manual and schematic, $55. A. R. Thornton, 6404 Overton Road, Louisville KY 40228.

TRS-80, Apple or PET users. 1000 games—utility—business application programs with total value of over $17,000 loaned to you free. Send $2 for program catalog, details kit & free newsletter. $2 refundable with your first free selection. Danules, 6342 N.W. 29 Ct., Margate FL 33063.

1802 Users—especially Elf II: Games! Galactica, Slot Machine, Craps I, Blackjack. All 4 for $10, cassette or memory dump. Blackjack occupies 0000-0900, others much less. Also custom programming, machine or Tiny BASIC. John Beringer, 2729 West Sahara Apt. 2, Las Vegas NV 89102.

TRS-80 cassette recorder (CTR-41) fix. 1. Eliminates gnd loop; 2. Gives control of fast-fwd & rewind with tone switch (no unplugging remote plug); 3. Auto level control on CLOAD & CSAVE (no adjusting volume on Lvl I or II). 72 hr svc (for cert. chk or MO), $15 incl. ret. postage in Cont. U.S. Ron Brandt, 201 W. McCoulskey, Terrell TX 75160. (214) 563-9790.

TRS-80 Assembler Disk Input/Output Programs. Assembler programs available for random or sequential files and blocked or spanned records. These programs make much more efficient usage of memory and machine capability especially in a business -oriented program. Each program only $1. Please specify amount of RAM memory in your system. Gerald Barker, Dept. of Computer Science, Okla. State University, Stillwater OK 74074.

TRS-80 Cheap Printer complete pkg. Includes source listings, Edtasm patches, and text proc., for Model 15 or other Baudot TTY. Incredibly simple interface. $25 on cass. SASE, for details. N. Gregory, 107 Fairdale, Lex. KY 40505.

For Sale: 2102-1 low power 450 ns static RAMs, 101 for $80; CSC Logic Probe (Model LPK-1) assm., $15. Paul Whitelock, 1922 Englewood Ave., Balto. MD 21207.

OSI 1P Owners: Want some important info not found in your Users Manuals plus other helpful info? Send $1+ SASE for a sample and list to: William J. Dirks, 44th USAAD, APO NY 09221.

H8 Owners! My powerful ROM monitor lets you examine and alter memory and registers, load hex tapes, execute programs with breakpoints, etc. from your console in addition to all front panel functions. Hex and octal versions available. For details, send self-addressed stamped envelope to: Patrick Swayne, 290 Springdale, Sebastopol CA 95472.

TRS-80 16K Level II programs on tape. Investment portfolio evaluation, $6. Itemized income tax deduction records, $5. Both for $8. Dick Carney, 1927 S. Dewey, Bartlesville OK 74003.

TRS-80 Level I or II—Horse Race, Alphanumeric Graphics Signboard, Stock Charting; $4.50 each. Printer listing; $1.50 each. Five popular games; Cassette $5.95 Option Strategies, Level II 16K; $9.95 (listing at $3 each). C. Zalnerunas, 3034 W. Columbus Ave., Chicago IL 60652.

PET Printer Adapter with built-in P/S and connectors, $90 complete. G. E. computer grade caps. 2900 uf, 350 V, $6.50; 850 uf, 450 V, $3.50. All new but surplus. J. Gatliff, PO Box 627, Eau Gallie FL 32935.

Database program for 16K TRS-80 Level II or DOS BASIC. Define the number of fields, save data on disk or tape. Display, search, edit, print and sort functions. $10. Antonio Zamora, 3879 Beulah Rd., Columbus OH 43224.

Must Sell . . . Need Money A fully populated OSI Challenger III System. Includes 32K, dual 8" floppies, keyboard, video, cassette, graphics, monitor, 16 serial I/O board, 2 8-bit D/As, 1 A/D with 8 channel multiplexer, 3 parallel I/Os, 1 megabyte memory management, software switchable address for all three micros, with connecting cables . . . fully operational. Will sell for $4500. Plus 5 free disks of software with purchase, includes BASIC, Assembler, Editor, CH III Op Sys, WP-1A, OSI Level I and much, much more. Pat Reynolds (713) 674-8367.

North Star BASIC Total Cross-Reference is a superb modify/debug aid for your big programs. $15 including disk. Free manual. Robert Siegenthaler, 2690 Gilham Road K1, Eugene OR 97401. (503) 687-0181.

Ohio Scientific C2-4P with connected RS-232 port and switch selectable baud rates, works perfectly. Cost $598: will sell for $498. Pat Reynolds (713) 674-8367.

TRS-80—PR-40 SWTPC Printer Interface Kit. All parts (including cable) and instructions. Installation time: 25 min. Price: $50. David Wichmann, PO Box 3204, Manhattan Beach CA 90254 (the PR-40 is an impact printer).

TRS-80 Printer Interface. An RS-232 printer interface which works off of cassette port number 2. May be used with or without the expansion interface. 110-9600 baud and other options. Excellent software support. Tape version, $39.95; disk version only $49.95. Robert Webster, E101 Cedar Oaks Apts., Stillwater OK 74074.

Payroll: Convert Radio Shack Level I payroll to Level II-Disk-Line Printer. Stores week after week with name and date print. Correction and change built in. Send Level I tapes and $15 to Jim Russell, 110 Beechmont Dr., Carmel IN 46032.

TRS-80 16K Level II JBLOCK—partitions Mem.—Have up to 8 prog. in mem. at once. Disassembler prog.—6K—Zilog mne. Listing $2 ea., cassette $3, extra. John B. Sweeney, 1 Victor Dr., Albany NY 12203.

Heath H8, 24K memory, cassette tape, extended BASIC. H9 video terminal. All documentation and software. Both up and running and checked by factory. Asking $1600 and I ship. Randy Humphreys, Box 250, Sudan TX 79371. (806) 227-2381.

Video Terminal—12 in. monitor and 64-key ASCII kybd in nice cabinet. All power inc. 12.5 Mhz. $140 Ppd. K. Lemkelde, 10658 Green Mtn. Cir., Columbia MD 21044. (301) 997-0634.

Apple II Software: Revive—Restores NEWed or wiped-out Applesoft programs, $8; Disc Copy—Automatically copies Applesoft or Integer BASIC programs from one disk to another with only one disk drive, $15; Applesoft Utility—has Renumber, Append, Hex/Dec-Dec/Hex Converters, Line counter, Byte Counter, Slow/Stop List and REM, PRINT and POKE statement formatters. Needs 20K without ROM card or 8K with. Jules H. Gilder, 2022-79th St., Brooklyn NY 11214.

SWTPC: Complete disassembled source listing of V2.0, V2.2 or V2.3 8K BASIC. Specify version $15. Mik Innova, PO Box 53, Huron OH 44839.

Radio Broadcast Log Program: TRS-80 Level II, 16K for use with up to 4 carousels. Separates competing accounts, counts minutes. D. H., Box 366, Hemingway SC 29554.

TRS-80 Reverse Video Mod. Have your display look like this ad (white background with black letters). Video display looks 100% better. For $2 you get complete details. No external parts needed, no software. Simple to do. Abel Hernandez, 214 N. Bond, Dallas TX 75211. (214) 330-6409. SASE.

TRS-80 original programs; business, games, educational, graphics; 50 for $25. LI or LII ppd in US. No COD. SASE for complete list. N. P. Jensen, 1589 Blossom Pk, Lakewood OH 44107.

For Sale: New factory-wired Meca Alpha-1 dual cassette. Includes Meca OS Ver. 3.0. Couldn't figure out how to use it with my system! Take advantage of my mistake. $600 (or make reasonable offer). Send certified check or money order, I'll pay shipping. W. D. Wilkens, 24 N. Third St., Womelsdorf PA 19567.

Mastermind for the basic Elf. On tape $5; listing $2. Alfred Pacheco, 2500 George Washington Way, Richland WA 99352.

TRS-80 Software by Dan. Super-Calculator: simulates programmable desk calculator. 7 mem. on screen, 100 data reg. printer control, $6.95. Code practice tape generator, $5.95. Letter-writer: word-process. at a micro-price, $8.95. Call or send for list. Dan Lauck, R.R. 1, Box 87, Sherman IL 62684. (217) 629-7693.

TRS-80 utilities. 1. Translate machine language programs to assembly language with Level II disassembler plus, $9. 2. Use editor -assembler with disk & disassembler & copy disk files to tape. 3. Machine-language programs execute much faster than BASIC—create your own without hassle—BASIC Compiler. SASE for information Atkwill, Box 40387, San Francisco CA 94140.

For Sale: Axiom 801P Line Printer. Used one month, went to larger unit. Standard parallel port interface. $275. A. H. Moser, Route 1, Box 16, Samuels ID 83862.

MMD-1 W/256 bytes RAM, all documentation, bugbooks 3&5. Excellent tutorial system. $250 + UPS. Frank Debolt, 114 Eastpines Rd., Savannah GA 31410. (912) 897-1384.

In the interest of accuracy and efficiency we ask that all classified ads please be typed, double spaced. Please do not send in ads in all caps; use upper and lowercase where appropriate. We generally will lowercase all uppercased words in the interest of space.

PUBLISHER'S REMARKS

(from page 5)

around, blinking their eyes and shuffling their tails, the game would have been a drag. The snakes occasionally got into arguments about the game, which helped too. The game, one mostly of chance, was of no consequence, so I figured that we might sell the program as perhaps half of a package, combining it with a better game having poorer graphics.

Splitting the royalties for the package would have resulted in perhaps a 60¢ royalty per sale for the programmer . . . figuring a $7.95 package. Some other publisher offered him a $3 royalty per sale, so he decided to go that route. I'll be interested to see where he ends up in the long run. I would expect that Instant Software would be able to sell between 50,000 and 75,000 copies of his program, bringing him a royalty of $30,000 to $45,000. The publisher he contracted with is bringing out this program at $15. One of the sad facts of life in the program-sales business is that people are no longer paying $15 for game programs. He'll be lucky to sell a few hundred copies at that price.

PET-POURRI

(from page 7)

good at first, but after a short time it seemed that the programs would "fade" away and no longer load. Other tapes seemed to be bad on the tape heads. After only a few saves and loads the tape heads needed to be cleaned to work reliably. Still others made awful noises while running.

A source of good-quality tape is Computer Way (PO Box 7006, Madison WI 53707). See last month's PET-pourri for details.

After you spend all the time making your tapes, I recommend neatly labeling them. Blank labels are inexpensive and available in several colors (including red, white, blue, green and yellow) for $3.45 per 100 from Computer Way. These labels come on sheets of 10 and include strips to use to label your box or cases. Computer Way also has a roll of 1000 white labels for $14.95.

There are several ways to organize your tapes for easy access. Demco Educational Corp. (2120 Fordem Ave., PO Box 7488, Madison WI 53707) has several good cassette storage systems. They have revolving round units that hold 25 cassettes and can be stacked on top of each other. Cost is $45 for three units. I have six of these and like them. They also have several cases with flip-out holders and several types of cassette books.

20th Century Plastics (3628 Crenshaw Blvd., Los Angeles CA 90016) has three-ring cassette binders for $6.95 each (three or more). These hold 12 tapes and allow you to store listings and documentation neatly. Radio Shack has a similar folder (without the three-ring binder part) for $2.59 each that holds 12 tapes.

Of course, you can store your cassettes in single boxes. There are three types of boxes generally available. A hard-plastic Norelco-style hinged box is good if you wish to see through the top, but these are a bit brittle. A translucent, soft, hinged box is more durable and usually a couple cents cheaper. There is also the two-part soft-plastic type. The first two are available from Computer Way.

Remember to keep your tape heads clean for reliable service. Computer Way has both cotton and lint-free foam-tipped swabs and special tape-head cleaning fluid.

A Final Note

In the May 1979 PET-pourri appeared "Tape Test" by Jim Butterfield (see p. 7 of the May issue). The original version had data loaded directly into memory. By converting this into DATA statements for listing, the data overrides these lines and confuses the PET. The machine-language data then has to be relocated to higher memory locations. Fortunately, Harvey Herman of Greensboro NC has done this for us. His line changes to "Tape Test" are:

```
180 SYS(2280):END
900 Z=2280
1004 DATA 2, 169, 80, 141, 25, 2, 169, 9, 141, 126
1010 DATA 206, 48, 204, 0, 32, 92, 9, 44, 64, 232
```

Send your discoveries and ideas to Len Lindsay, 1929 Northport Dr., Room 6, Madison WI 53704.

Another Hexadecimal Keyboard

This could be the key to what you've been looking for.

O. C. Stafford
3702 Holts Chapel Rd.
Greensboro NC 27401

If you have been looking for a simple, concise, 8-bit hexadecimal keyboard, the unit shown in the photo should interest you. It uses the National Semiconductor 74C922 keyboard chip as the encoder.

The circuit diagram is shown in Fig. 1. The feature that makes this circuit work is the data-available pulse, which comes true before the data changes on the output pins. With a type D flip-flop, the data is transferred to the flip-flop on the positive-going edge of the data-available pulse. Two successive operations of the keys cause a full eight bits to appear on the output lines.

There are three extra keys on the keyboard: one for the input switch and the other two for putting my 1802 CPU in the Load or Reset mode. Construction is facilitated through the use of a PC board. The full-size pattern is shown in Fig. 2. Fig. 3 shows the parts layout. The key caps should be arranged as shown on the PC board legend and the photo. ■

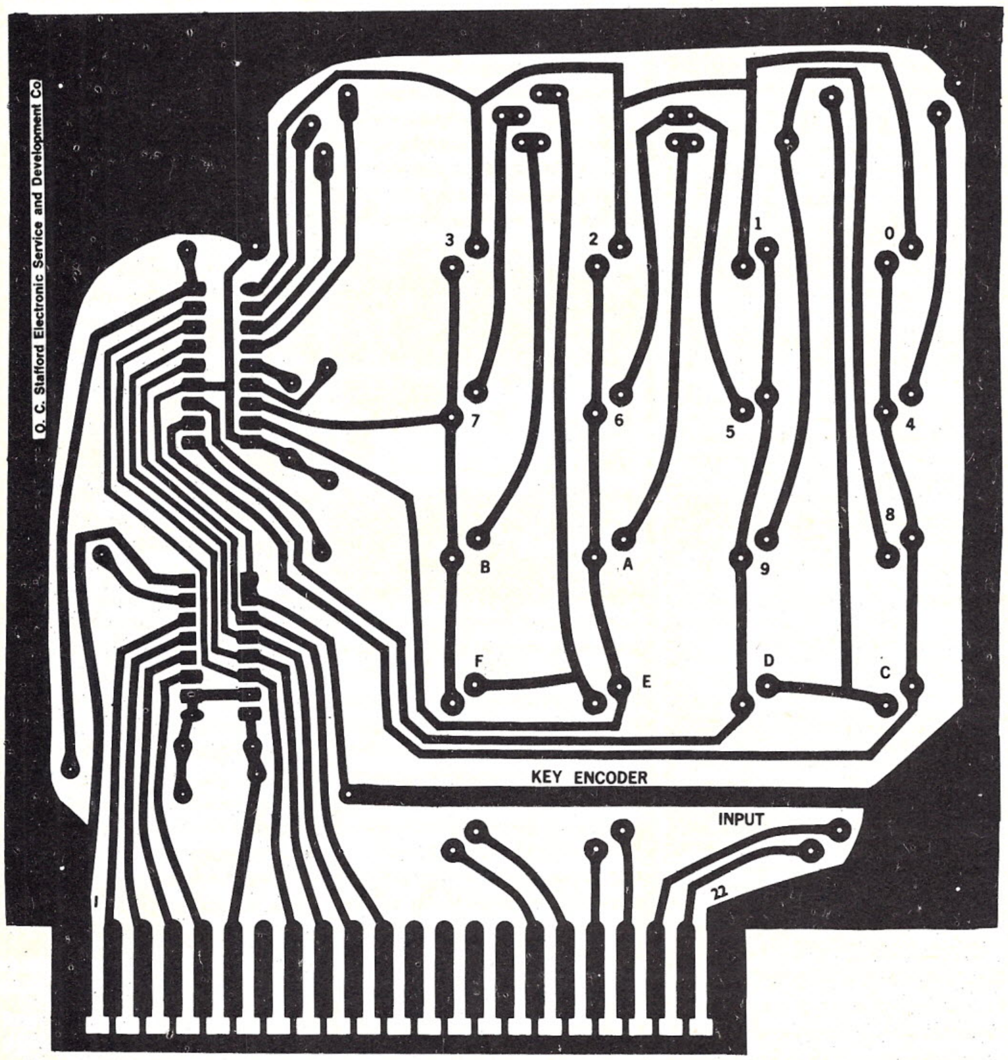


Fig. 2. Full-size PC board pattern for the hexadecimal keyboard.

1-74C922 keyboard encoder IC1	
1-74C175 type D flip-flop IC2	
1-0.15 disk cap. C1	
1-5.0 uF electro C2	
2-0.1 uF disk C3, C4	
1-keyboard hexadecimal K-1	
19-keypad unencoded	
James Electronics	
1021 Howard Ave.	
San Carlos CA 94070	
1-18-pin socket	
1-16-pin socket	
1-card-edge connector	
1-PC board	
O.C. Stafford Electronics	
427 S. Benbow Road	
Greensboro NC 27401	
Parts kit (less PC board)	$28.50
PC board drilled	5.80
undrilled	3.50
negative	2.00

Parts list.

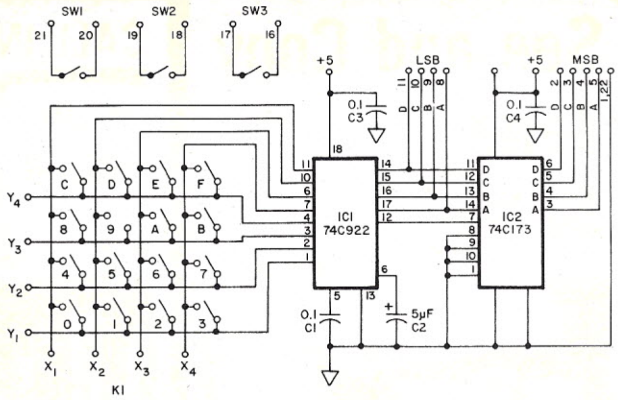
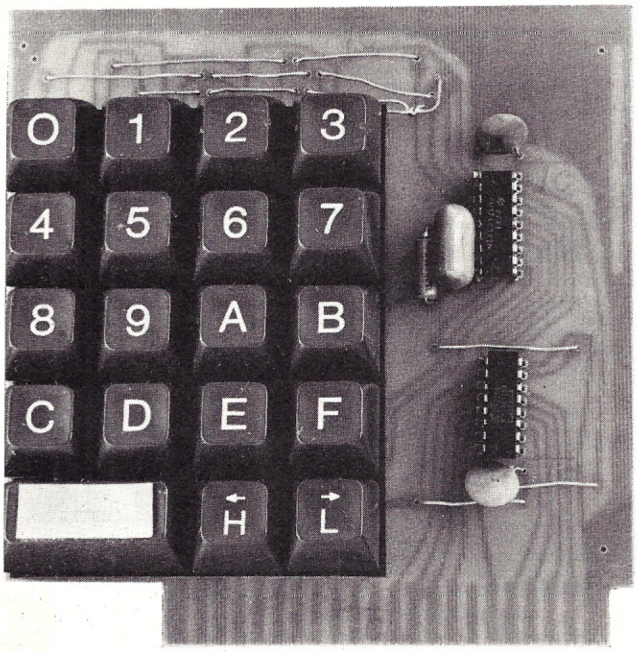


Fig. 1. Schematic for the hexadecimal keyboard.



Hexadecimal keyboard. Key covers should be arranged as shown.

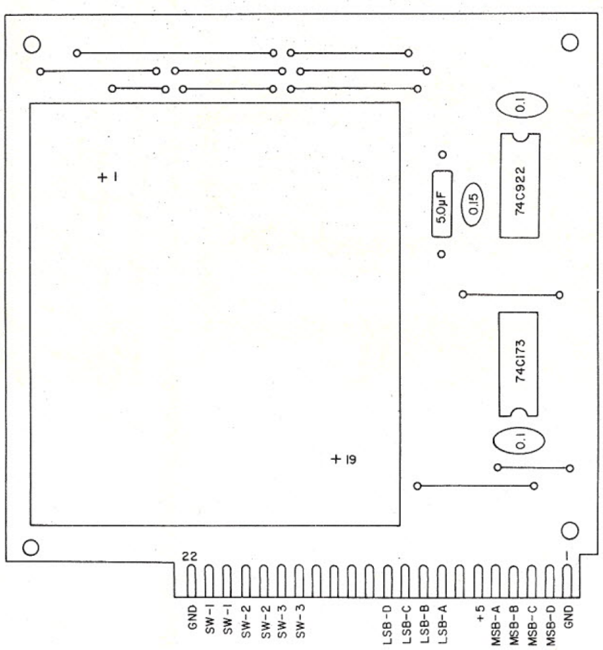


Fig. 3. Parts layout for the circuit board pattern shown in Fig. 2.

Dear H8 Owner:

For over a year, we have been supplying top-quality programs written especially for the Heathkit H8. The satisfaction expressed by our customers in regard to both our products and our service has been especially gratifying. It is with a great deal of pride, therefore, that we present our products here. We trust you will take the time to read about them.

PERSONAL USE PROGRAMS:
Several personal use programs are available. PERSONAL BUDGET is a sophisticated personal budgeting/accounting system. INTEREST performs simple interest, compound interest, and mortgage amortization calculations. MENU produces nutritious, within-calorie-limits menus. SHOPPER arranges your grocery list in the shelf order of your favorite store. All four programs are supplied on one diskette. Specify Ext. B.H. BASIC or Microsoft BASIC. (No. 111.1, $30)

MAILING LIST/INFORMATION RETRIEVAL PROGRAMS.
Designed primarily as mailing list programs, these three disk-based products can also be used for storage and retrieval of other information. MAILIST (No. 114.1, $30) runs under Ext. B.H. BASIC and is extremely flexible. The number of items to be recorded about each entry is specified by the user. Sorting and retrieval can be done on any field of the user's choice. Printout can be done to any device and on various sizes and layouts of labels, as well as in the form of individually addressed letters. MAXI-LIST (No. 116.1, $30) also runs under Ext. B.H. BASIC, but utilizes the full capacity of available disk space rather than being limited to the number of names that can be loaded into memory at once. Automatically maintaining a backup file, its other features are the same as Mailist except that sorting must be done on one field only. MAIL-PRO (No. 127.1, $30) runs under Microsoft BASIC. Its features are similar to those above but the program was still under development when this ad was composed.

ACCOUNTING PROGRAMS.
Our growing family of accounting programs now includes two general ledger packages and a fixed assets package, all on disk. GENERAL LEDGER (115.1) runs under Ext. B.H. BASIC and provides a complete double entry accounting system with error-checking routines and comprehensive printouts. Account numbers and names are determined by the user. GENERAL LEDGER II (125.1) runs under Microsoft BASIC and provides a more sophisticated system, complete with automatic bank reconciliation and automatic closing. FIXED ASSETS (128.1) maintains complete records on fixed assets and automatically computes depreciation, using the method chosen by the user for each asset. Running under Microsoft BASIC, Fixed Assets can provide data to General Ledger II. Exact prices on the accounting programs had not been set when this ad was composed; therefore, please write for additional information. Brochures are free; the complete user's manuals are $5 each, which may be credited toward later purchase.

CASSETTE PROGRAMS—LAST CALL.
As long as our current stocks last, our cassette-based programs are available at $8 per package. Packages available are #101.1 Games (formerly $10), #102.1 Personal Finance (formerly $10), #103.1 Cook's Helper (formerly $10), and #104.1 Mailing List (formerly $20).

Now that you've read about our programs, we hope you will join our growing family of satisfied customers by completing the order form below. Shipping is free to all U.S. locations.

Sincerely,

JIM CLARK

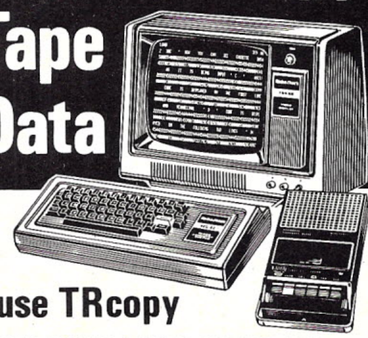
Please send the following: ________ Price ________ Total Amount Enclosed: $ ________

NAME ________
ADDRESS ________
CITY, STATE, ZIP ________

E45

Ed-Pro Incorporated
P. O. Box 490156 — Atlanta, Ga. 30349

See and Copy Tape Data



use TRcopy

WITH YOUR LEVEL II TRS-80*

TRcopy is a cassette tape copying system that lets you SEE what your computer is reading.

COPY ANY CASSETTE TAPE**
With the TRcopy system you can copy any TRS-80 Level II cassette tape whether it is coded in Basic or in machine language. You can also copy data created by programs and you can copy assembler listings.

YOU CAN SEE THE DATA
As the tape is being loaded, you can SEE the actual data byte-for-byte from the beginning to the end of the program. Up to 320 bytes are displayed at one time. ASCII characters are displayed on the first line and hexadecimal code is displayed on the following two lines. Data is displayed exactly as it is input including memory locations and check sums.

IDENTIFY PROGRAMS
With TRcopy you can identify programs on cassette tapes without written documentation because you can SEE the filename. If you forget to label a tape, you can use TRcopy to display the tape contents and identify the cassette.

VERIFY CASSETTE TAPES
With TRcopy you can verify both the original tape and the tape copies. You can make certain that your machine reads the original tape correctly and that it makes byte-for-byte copies. TRcopy also counts as it reads giving you the exact length of the data.

MAKE BACKUPS FOR YOUR PROGRAMS
Now you can make backup copies of your valuable programs. Many times a cassette that you make will load better than one that is mass produced. The original can then be kept as a backup in case the copy is damaged.

MAKE COPIES OF YOUR SOFTWARE
If you are in the software business you can use TRcopy to make tested copies of your programs for sales distribution. TRcopy produces machine language tapes that are more efficient than those produced by the assembler itself.

RECOVER FAULTY DATA
With TRcopy you can experiment with the volume and level controls and you can SEE what the computer is reading--even if your computer will not read the data through normal read instructions! In this way it is possible to read and copy faulty tapes by adjusting the volume control until you SEE that the data is input properly.

SIMPLE – FASCINATING – FUN
TRcopy is not only a practical utility program. It is also a fascinating graphics program that lets you SEE, for the first time, cassette data as your computer is reading it. And it's as simple as 1-2-3. Just load, verify and copy. You will now be able to use cassette tapes with confidence knowing that TRcopy is there when you need it.

The TRcopy system is a machine language program with documentation explaining tape leaders, sync bytes, check sums and other formatting conventions. With the TRcopy system, you can SEE what you are doing!

TRcopy System Including Cassette Tape and Documentation **39.95** POST PAID

Orders accompanied by money order or cashier's check mailed same day. Orders paid by other check shipped in 14 days. No COD's. Return within 10 days for a full refund if you are not satisfied.

N.D. Orders Add 3% Sales Tax. *TRS-80 is a trademark of the Tandy Corporation. **You cannot copy the TRcopy cassette.

ORDER FROM

Data/Print
DEPT. KB, BOX 903, FARGO, N.D. 58107

CALL IN YOUR ORDER Now!

Call Toll-Free 24 Hours

If you have a Master Charge or a Visa credit card, you can call toll-free from the nearest telephone and have your TRcopy system on its way to you today.

Call 1-800-437-4144 anytime – 24 hours a day.

For calls from N.D., Hawaii or Alaska call collect 1-701-237-0216

CALL RIGHT NOW!

VISA · master charge

NOW USE TRCOPY WITH YOUR PRINTER

Included at no extra cost – Now you can use the TRcopy system to output tape data to a line printer or a quick printer. The data is printed exactly as it is input from the tape including file names, memory locations and check sums. A printed copy can be especially helpful in the analysis or recovery of records contained in tape data files.

ORDER YOUR TRCOPY SYSTEM NOW!

D50

TOLL FREE 24 Hours SAME DAY SHIPMENT

✓ Reader Service—see page 179

NEW & IMPORTANT for TRS-80

DIAGNOSTIC SOFTWARE
for TRS-80 level II

Comprehensive circuit analysis for your system
- RAM—tests for Random Access Memory errors
- ROM—tests for Read Only Memory errors
- CPU—tests for errors in Processor functions
- I/O—tests for Input/Output errors in peripheral equipment

The one "necessity" for any TRS-80 system—use as preventive maintenance for detecting circuit malfunctions and as a diagnostic tool for pin-pointing hardware problems.

$34.95
(specify tape or disk)

Business Software for TRS-80

Business Inventory - Level II - 32K - 2 Disk Drives **$49.95**
● Random Access - Double Precision - 650 Item ● Reports Printed ● Inventory ● Activity to Date ● Re-order List ● On-order List ● Out of Stock List ● Dept/Sales Report ● Inventory Worksheets

Comprehensive Business Data Base - Level II - 32K - 2 Disk Drives **$129.50**
● Account Number Generation (Check digit) ● Daily Transaction & Trail Balance ● Statement of Accounts & Aging Report ● Inventory Inquiry ● Invoicing

Avail. soon for Data Base - General Ledger, P/L, Tax Reports

Payroll (115 employees) Level II, Disk, 32K **$49.95**
● Month, Qtr., Year to date reports ● Random Access ● Double Precision ● Can change individual fields ● Prints Checks & Stubs

VR Data corp. V19 *Folcroft Industrial Park • Folcroft, PA 19032 • (215) 461-5300*
MORE DATA PER DOLLAR

master charge — VISA

SPACE WAR!!

TRS-80* APPLE II*

Real time graphic SPACE WAR like that played on large computers is availible for the TRS-80 and the APPLE II in machine language. Two players combat in space while orbiting the sun in individual space ships. Each ship can rotate left or right, accelerate, fire up to five missiles at a time, and enter hyperspace. The program has five levels of play, for beginner to advanced, as follows:

MODE:		
(1)	BEGINNER	(No floating - ships move like tanks)
(2)	INTERMEDIATE	(Floating - like real space ships)
(3)	ADVANCED	(Adds Sun's gravity)
(4)	NO EDGE BOUNCE	(Ships can move off screen edge)
(5)	REVERSE GRAVITY	(Sun pushes ships and missiles away)

Availible for 16K TRS-80 Level I and Level II or 16K APPLE II. TRS-80 version requires no additional hardware due to keyboard design. APPLE II version requires hi-resolution graphics and two four-button consoles - complete plans for building your own are included.

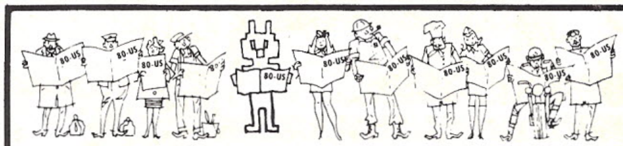
Please specify TRS-80 or APPLE II.
Send $9.95 check or money order for cassette tape and manual.

All orders are postage paid.

Texas residents add 5% sales tax.
*TRS-80 is a trademark of Radio Shack, a Tandy Company.
*APPLE II is a trademark of APPLE Computers.

DOG
DEVICE ORIENTED GAMES, INC.
P.O. Box 400727
Dallas, TX 75240 D52

$9.95



YOU'VE READ THEM ALL-
NOW GET THE BEST!

The 80-US Journal

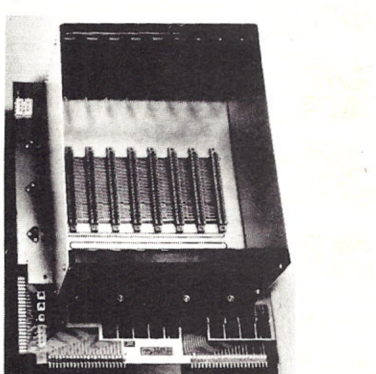
A PUBLICATION FOR TRS-80 USERS - COVERS **ALL** TRS-80 CAPABILITY!

BI-MONTHLY, SEND $16.00 CHECK OR MONEY ORDER FOR A 1 YEAR SUBSCRIPTION TO:

80 US E37
PO Box 7112
TACOMA, WA. 98407

MASTER CHARGE, VISA OK - (206) 759-9642
WANT TO CHECK IT OUT FIRST? SEND $3.00 for current sample copy.

A SELECT CHOICE OF SOFTWARE IS AVAILABLE



PRICE: $129.00
We also carry the **SYM-1** Microcomputer with manuals **$269.00**

VAK-1 MOTHERBOARD

- Designed specifically for use with the **AIM-65**, **SYM-1**, and **KIM-1** microcomputers
- Standard **KIM-4*** Bus
- Fully buffered Address and Data Bus
- Provides 8 expansion board slots
- Complete with rigid card-cage
- All IC's are socketed
- Provides separate jacks for one audio-cassette, TTY, and Power
- Completely assembled (except for card-cage)

We manufacture a complete line of high quality expansion boards. Use reader service card to be added to our mailing list, or U.S. residents send $1.00 (International send $3.00 U.S.) for airmail delivery of our complete catalog.
**Product of MOS Technology*

R20
RNB ENTERPRISES INCORPORATED
2967 W. Fairmount Avenue • Phoenix, AZ 85017 • (602) 265-7564

✓ *Reader Service—see page 179*

SPOOLER for NORTHSTAR

Run Programs and Print Spooled Files at the Same Time

The MASP-SPOOLER allows PRINTOUTS and LISTINGS to be diverted to named DISK FILES. Under command, these files can be printed CONCURRENT with a running BASIC program. MASP is completely transparent to DOS and BASIC. It requires 4K at 0000, 8000, 9000, or A000 **$69.95**

WORD PROCESSOR

This program features full justification, centering, tabs, underlining, paging and more. It is written in BASIC so that it can be easily modified. Text files are written using the BASIC editor, so you do not have to learn a new system. It outputs at least 120 wpm with a 30 cps printer. **$49.95**

DATA BASE and MAILING LIST

Structured as a mailing list, this program provides access to data at high speed using fast sorting and search techniques. All fields for each record are variable length so almost any type of data can be stored. Fields include: LAST NAME, FIRST NAME, FAMILY MEMBERS, COMPANY, ADDRESS, PHONE; and 6 more you define. Current balance and accounting data is also kept so the system functions as an accounts receivable package. You can also search on any field. **$129.95**

Mini manuals for any system available for $3.50 each

MICRO WORLD
COMPUTER STORE INC. M61
435 Main Street Johnson City, N.Y. 13790

: PET and TRS-80 :

'Just LOAD and GO' Software
Pre-recorded Business Programs
USEFUL-PRACTICAL-LOW-COST!
NO PROGRAMMING EXPERIENCE REQUIRED!

- **GENERAL LEDGER**—For home businesses, sole proprietorships, small corporations—$19.95 plus $1.50 s&h, requires 8K min. user memory
- **CHECKING ACCOUNT**—For personal bank accounts—$19.95 plus $1.50 s&h, requires 8K min. user memory
- **RENT ACCOUNTS**—Records on rental properties—$16.95 plus $1.50 s&h
- **LEGAL DIARY**—For Attorneys (Client Accounts) $16.95 plus $1.50 s&h
- **TRUST ACCOUNTS**—For Attorneys (Client Accounts) $16.95 plus $1.50 s&h

Programs include 2 PerCom "Pilon-30" record cassettes—money back guarantee

HUSTLER Series 1 for PET(tm) are now available in Britain and Europe through:

PETSOFT
PO Box 9, Newbury Berks. RG13 1PB, England

Specify which computer is used.

Dealer Inquiries Invited

All mail orders must be pre-paid.

c/o Kenneth Harris **COMPUTERS ONE** C81
1314 South King St. Honolulu, HI 96814
Suite 1156 808-536-7735

Expand your TRS-80. Save $100.

Meet the Vista V80 Mini Disk System. The perfect way to upgrade your TRS-80* system. Inexpensively. (Our $395.00 price is about $100.00 less than the manufacturer's equivalent.) Here's how it can help you.

23% more storage capacity. Useable storage is increased from 55,000 to 65,000 bytes on drive one.

8 times faster. While electronically equal to the TRS-80 Mini-Disk system, track-to-track access is 5ms versus 40ms for the TRS-80.

Better Warranty. The V80 carries a 120 day warranty – longer than any comparable unit warranty available.

The Vista V80 Mini Disk System comes complete with Minifloppy disk drive, power supply, regulator board and case. And it's ready to run – simply take it out of the box, plug it in and you're ready to go.

Vista V20

1320 E. St. Andrews Place, I
Santa Ana, CA 92705
(714) 558-8813

At Vista, we mean business

*TRS-80 ©Tandy Corp.

1802

EXTENDED MONITOR

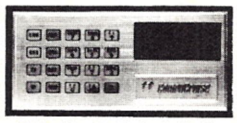
- Self relocating to any page boundary.
- Requires 1K of RAM or ROM plus stack.
- User may design his own I/O routines.
- 25-page user manual with hex listing.
- Supports user relocatable programs.
- Kansas City cassette routines.
  No tone leaders or trailers needed.
  Simple interface requirements.
  Check sum error detection.
  Enables sequential file processing.
- ASCII to hex and hex to ASCII routines.
- Print character strings, registers, memory.
- Examine/alter memory, load paper tape.
- Much more.

OP1802-1 K.C. tape*************** $20.00
OP1802-2 Paper tape************** $18.00

BENCHMARK COMPUTER SYSTEMS B51

17 Hanover Place
Hicksville, L.I., N.Y. 11801

CAR COMPUTER



$184.00 w/cruise
$149.00 wo/cruise

Shipping included. Calif. res. add 6% sales tax

- **clock-12 quartz, stopwatch & alarm**
- **elapsed time-trip time-time to arrival**
- **trip distance-time to arival-to empty**
- **distance traveled & gas used since filling**
- **fuel-to empty-to arrival-used on trip**
- **mpg or gph-current or average for trip**
- **cruise control, current speed or preset**
- **converts figures to metric if desired**
- **speed-current or trip average**
- **temperature-inside-outside**
- **bright fluorescent display & much more**

Expiration Date ____________ Charge on my ☐ B of A Card ☐ Mastercharge

Card # ____________

Signature ____________

Computer Forum

- BUSINESS
- EDUCATIONAL
- PERSONAL

14052 East Firestone Blvd. • Santa Fe Springs, CA 90670
(213) 921-2111 • (714) 739-0711

TRS-80 COMPUTING

non profit newsletter

12 Issues For $15.00

(US)

and now

PEOPLE'S SOFTWARE

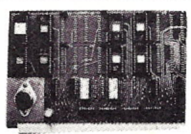
at popular prices

- $7.50 per tape plus 50¢ postage & handling (CA residents add 45¢ tax)
- Tape 1: 42 Level II or 24 Level I (indicate which you want) business, home, educational
- Tape 2: 76 programs from "Common Basic Problems" by Osborne Associates
- Write for free TRS-80 bulletin

CIE COMPUTER INFORMATION EXCHANGE

Box 158 San Luis Rey, CA 92068

NEW 32K EPROM/RAM MEMORY CARD



SS-50 BUS
5V only

DSD P/R-32K........... $27.00
2716 EPROMS & TMS 4016 2K×8 Static RAM. 4 Independent 8K memory blocks. Size 9" × 5½"

DSD 2114-16K.......... $27.00
16K Static RAM memory card using the 2114 or TMS 4045 1K×4 Static RAMS. Size 9" × 5½". 2 Independent 8K memory blocks. 5V Only.

NEW DSD U P 8255 (C&M)
.... $14.00
Universal parallel interface card for Both the SWTP 30 pin I/O BUS (M) and the DSD 27 pin BUS (C). Utilizes INTEL's 8255 PROGRAMMABLE I/O chip. Wire wrap area for full UTILIZATION of the 8255. Card sizes 5½" × 5" (M) 5½" × 4½" (C).

All cards are bare board with edge connectors and Data.

DIGITAL SERVICE & DESIGN
P.O. BOX 741
NEWARK, OHIO 43055 D53

Ohio residents add 4.5% sales tax.

TRS-80 HARDWARE

80's COUPLER—COMMUNICATION and SOUND for LEVEL I, II, DISK SYSTEMS. Hardware device that allows 2 TRS-80's to send and receive data in BASIC or Machine Language via the cassette ports. Simply insert recorder plugs into the 80's coupler and use INPUT #-1, PRINT #-1, CSAVE, and CLOAD, or use machine language for fast data transfer between machines. A tape of games using the coupler is provided. Kit with all parts is $34.95; completely assembled, $44.95.

Option—switch selectable speaker which allows the use of sound from your TRS-80. Can be used with MICRO-MUSIC to create your own orchestra. Add $5.00.

SOFTWARE Disk Software

Multiple Regression 2.0—menu orientation for ease of use, save and recall data from disk, user specified transformations, x-y plots, formatted for screen or printer, complete data editing, complete model specification ability, least squares estimates, t-values, standard deviation of estimates, variance-covariance matrix, correlation matrix, ANOVA table, R-squared, Durbin-Watson Statistic, Residual plot, Residual print. $45.00

Eigen values and vectors—determines the eigen values and vectors of a symmetric matrix. $39.00

Text Editor—blinking cursor, line insert/delete, character insert/delete, repeat key, disk I/O plus other features. $34.95.

LEVEL II SOFTWARE

Differential Equations—Solutions to 1st and 2nd order differential equations. Six methods are provided. All relevant intermediate calculations are printed. Excellent for educational purposes. $29.95.

Matrix Inversion Subroutine. $9.95.

Multiple Regression 1.0, Linear Programming 1.0, Transportation Algorithm, Zero-One Programming, Heuristic Line Balancing, all $29.95 each

Stat Pack—Median, Mode, Mean (average, harmonic, geometric), Variance, Histograms, Tests (t, X^2, F), one variable regression, one and two way ANOVA $9.95.

Queing Statistics—Computes queing statistics for several M/M/1 and M/M/C models including finite population and finite que size. $9.95.

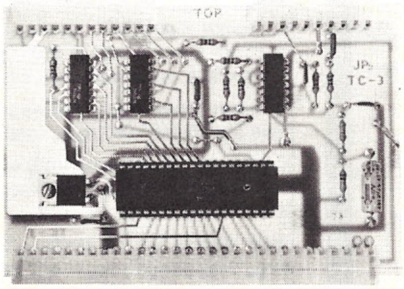
LOWER CASE MOD—Includes excellent documentation and all parts (absolutely nothing else to buy) compatible with the Electric Pencil. $14.95.

Add $2.00 for shipping and handling on all hardware orders
Foreign orders add $5.00 for shipping
SC residents add 4% sales tax

Quant Systems Q8

p.o. box 628
charleston sc
29402

Reader Service—see page 179



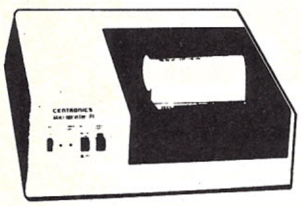
✓ *Reader Service—see page 179*

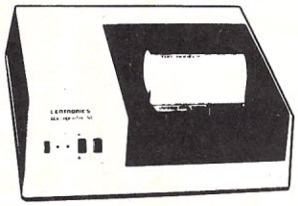
ADVANCED COMPUTER PRODUCTS

LOW COST HARD COPY

CENTRONICS®

Features:
150 Lines per Minute, 96 Characters – Upper and Lower Case, 5/10/20 Characters/Inch – Software Selectable, Elongated Characters, Underlining, Simplified Operation, Quiet, Audio Alarm, Long Life – Only 4 Moving Parts, Small Size and Light Weight, No Toners/Ribbons Required.





MODEL P1 W/PARALLEL INTERFACE

7-Bit ASCII, TTL Compatible Input/Output Signals, Compatible with Centronics and other Printer Product Lines.

List Price $495.00

ACP Sale Price $395.00

MODEL S1 W/SERIAL (RS232) INTERFACE

7 Bit ASCII, Serial RS232C Interface with Parity Selection, Switch-Selectable Baud Rate: 50 to 9600, 192 Character FIFO Buffer.

List Price $595.00

ACP Sale Price $525.00

Paper (6/case) ... $2.95/roll

ADVANCED COMPUTER PRODUCTS
P.O. Box 17329 • Irvine, California 92713 ✓A38
New Phone (714) 558-8813 • TWX: 910-595-1565

MICRODESK©

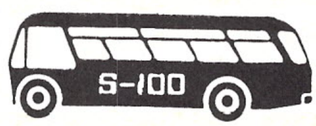
The perfect home for a personal or business computer

MICRODESK $159.50

48" x 24" x 26", Shelves 23" x 23", One adjustable.
See it at your computer store or order it from:

COMPUTER SYSTEMS DESIGN
906 N. Main Wichita, Kansas 67214
DEALER INQUIRIES INVITED

CATCH THE S-100 BUS



	LIST PRICE	OUR SPECIAL CASH PRICE
Imsai "8080"	750.00	625.00
Imsai MIO (two parallel, one serial, one control & Tarbell tape cassette port	195.00	150.00
SSM 5B-1 Music Synthesizer Board Kit	150.00	125.00
IMC PRO-1 Dual Extender Card	39.00	33.00
Thinker Toys Discus II - Double Density Disk System	1149.00	977.00
S.D. Systems Versafloppy Disk Controller Kit	159.00	135.00

	LIST PRICE	OUR SPECIAL CASH PRICE
S.D. Systems 32/64K Memory Board Kit	141.00	120.00
Godbout 8K Static Memory Board "Unkit"	139.00	118.00
Ithaca Audio TRS-80 Memory Upgrade Kit	119.00	98.00
Georgia Magnetics diskettes 8" soft sectored per box of 10	45.00	30.00
Vector Graphic 16K 250NS Static Memory Board Assembled & Tested	485.00	400.00

Check Our Prices On:
Cromemco, Vector Graphic, Mullen, Digital Systems, Centronics, Sanyo, Hazeltine, Computalker, IMSAI, ECT, IMC, North Star, Trace Plus Most Other Major Lines.

Subject to Available Quantities. Prices Quoted Include Cash Discounts. Shipping and Insurance Extra.

Bus... S-100, inc. ✓S60
Address... **7 White Place**
Clark, N.J. 07066
Interface... **201-382-1318**

✓ *Reader Service—see page 179*

NOTE to Kilobaud Microcomputing Subscribers

From time to time Kilobaud Microcomputing makes its subscriber lists available to carefully-screened companies and organizations whose products, services or information may be of interest to you. In every case, we must approve all organizations wishing to mail to our subscribers. In every case they receive a list of names and addresses only—no other information we may have is divulged. Each organization may use the list only once and agrees never to make any personal or telephone solicitations from it.

The overwhelming majority of our subscribers appreciate this controlled use of our mailing lists—it helps them shop conveniently by mail for products and services they need, often at substantial savings. A few people prefer their names not be used. It is to them we address this message.

If you do *not* wish to be a part of this service, please complete the form below . . . your name will not be used in this manner for any reason.

(If you asked us in the past to remove your name from our lists, there is no need to repeat the request.) Please allow about six weeks for your request to take effect.

☐ I wish to have my name removed from the list of subscribers receiving mail other than the regular subscription to Kilobaud Microcomputing.

Mr.
Mrs
Send to: Ms. ______________________
Please Print

Address ______________________

City ______________ State ________ Zip ________

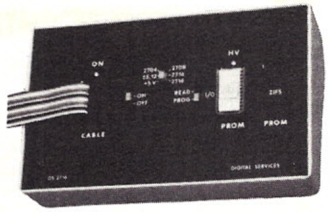
Mail this form with your mailing label from the latest issue (or fill out the information as it appears on the label) to:

Kilobaud Microcomputing
Subscription Services Dept.
P.O. Box 997
Farmingdale NY 11737

✓ Reader Service—see page 179

Reader Service—see page 179

DS 2716 EPROM PROGRAMMER



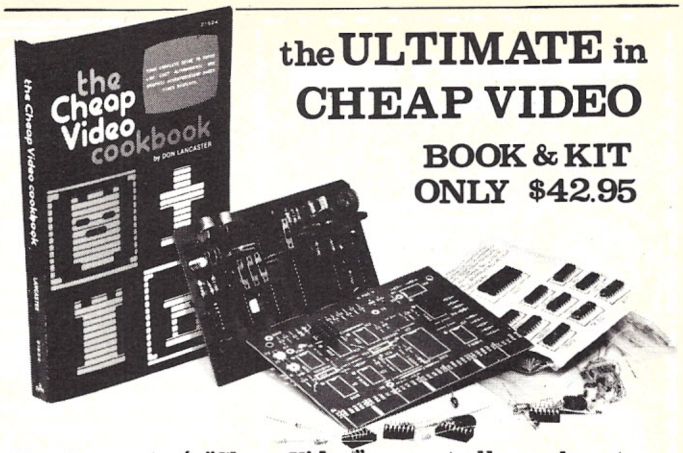
2758
2708
2716
2732
2532
I/O

- DS 2716 EPROM Programmer plugs directly into EPROM socket on any computer via 48" flat ribbon cable, or connect to 12 bits I/O port
- No other wires to hook to interrupt request, wait line or external power supply
- All control sequences, timing, and voltage (DC to DC converter) generated on programmer board
- Unique handshake control assures error free programming — short subroutine sends address and data over the address lines, and verifies latched data via data lines before programming pulse is applied
- Programs 2704, 2708, 2758, 2516 (switch selectable), 2532 and 2732 (jumper selectable)
- Program, Read, and I/O mode (switch selectable)
- DS 2716 can be used as a powerful 12 bit output — 8 bit input port

- **TRS 80 OWNERS — DS 2716 plugs into empty 2716-E socket in your computer (Level II, Level I single ROM) and you get EPROM PROGRAMMER and REAL WORLD INTERFACE (I/O ports)**

DIGITAL SERVICES ✓D55
P. O. BOX 21
KENT, OHIO 44240

DS 1716 Programmer
A & T $ 195.00
ZIFS (zero insertion force socket) add $ 7.00



the ULTIMATE in CHEAP VIDEO

BOOK & KIT ONLY $42.95

Don Lancaster's "Cheap Video" concept allows almost unlimited options, including:

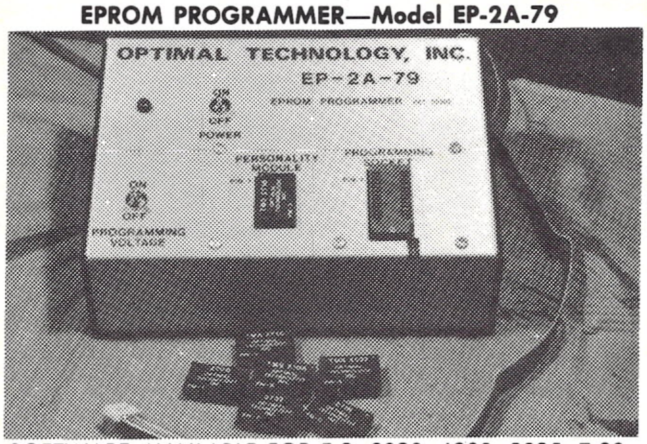
- **Scrolling· Full performance cursor.**
- **Line/Character formats of 16/32, 24/80, 32/64.... or almost anything.**
- **Graphics-up to 256 X 256 B&W; 96 X 128 COLOR (requires low-cost option modules)**
- **Works with 6502, 6800 and other micros.**

SPECIAL OFFER: Buy the Kit (upper case alphanumeric option included) & get the Book at 1/2 price.

✓P9 PAiA ELECTRONICS, DEPT.7-K, 1020 W. WILSHIRE BLVD., OKLAHOMA CITY, OK 73116

I'm Sold, PLEASE RUSH..... ( ) SEND FREE CATALOG
( ) TVT-6 5/8 Kit & Cheap Video Cookbook - $42.95 (enclosed)
( ) TVT-6 5/8 Kit only (book required for assembly)-$39.95
name: ____________
address: ____________
city: ________ state: ________ zip: ________
PAiA ELECTRONICS DEPT. 7-K, 1020 W. WILSHIRE BLVD., OKLAHOMA CITY, OK 73116

EPROM PROGRAMMER—Model EP-2A-79



SOFTWARE AVAILABLE FOR F-8, 8080, 6800, 8085, Z-80, 6502, KIM-1, 1802, 2650.
EPROM type is selected by a personality module which plugs into the front of the programmer. Power requirements are 115 VAC, 50/60 HZ at 15 watts. It is supplied with a 36 inch ribbon cable for connecting to microcomputer. Requires 1 ½ I/O ports. Priced at $145 with one set of software, personality modules are shown below.

Part No.	Programs	Price
PM-0	TMS 2708	$15.00
PM-1	2704, 2708	15.00
PM-2	2732	25.00
PM-3	TMS 2716	15.00
PM-4	TMS 2532	25.00
PM-5	TMS 2516, 2716, 2758	15.00

Optimal Technology, Inc. ✓O10
Blue Wood 127, Earlysville, VA 22936
Phone 804-973-5482

G2 PROGRAM LIBRARY

from

CASCADE ENTERPRISES ✓C121

ON CASSETTE TAPE

- If you use Solos™ or have an 8080 Sol™ System, G2™ Extended Basic is faster and better than others requires (15.5K) $49.95
- For our 6800 users G2™ Standard Basic specifically for the S.W.T.P.C.™ 6800 requires (7K) $34.95 Both G2 basics are by Microsoft™
- We have the following games and entertainment which will run in these machines:

THE GAME TAPES ARE ONLY $14.95 each	SOL TM	SORCERER	PET TM	TRS•80 TM	APPLE TM	SWTPC TM
G2 BEAT THE HOUSE	*	*	*	*	*	*
G2 CLINIC	*	*	*	*	*	*
G2 OUTWIT	*	*	*	*	*	
G2 THE MARKET		*	*	*	*	
G2 ADVENTURE				*	*	
G2 BLOCKADE			*	*	*	
G2 OIL TYCOON				*	*	
G2 PILOT					*	

G2 Beat the House includes Blackjack, Craps, Roulette and Slot Machine. Craps is a very big program. Ours left 3K usable in Level II 16K TRS-80™. Allows a family to play as a group against the computer.
Call us anytime except Sunday (916) 926-5154 or write

As an introductory offer—the first 500 orders get a free 10 minute blank cassette.

CASCADE ENTERPRISES
Box 213
Mount Shasta, CA 96067

Cal. Res. add 6% tax
Bank Cards add shipping & handling $2.00

☐ M/C ☐ B/A-Visa ________ Expires ________

Signature ________________

PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001

Programmer's Guides for the PET™

TIS

WORKBOOKS FOR THE COMMODORE PET 2001

Getting Started with Your PET WB-1 $4 00
Covers the fundamentals of PET BASIC calculator and program mode, data input and output, data representation, program storage on the cassette

PET String and Array Handling WB-2 $3 95
Covers string and substring search, concatenation replacement and manipulation

PET Graphics WB-3 $4 95
Covers use of cursor control and special graphics symbols to draw plots, histograms, and sketches

PET Cassette I/O WB-4 $4 95
Covers OPEN, CLOSE, string and numeric data files

Miscellaneous PET Features WB-5 $3 95
Covers the clock, random number generator, upper and lowercase alphabetic characters, saving memory space, etc

PET Control and Logic WB-6 $3 95
Covers IF, GOSUB, logical operations, and ON X

Add $1 50 for shipping and handling

Dealer Inquiries Invited

TIS
P.O. Box 921
Los Alamos, NM 87544

Money back guarantee

We also sell PET Software Write for details ✓T41

PET is a trademark of Commodore Business Machines

PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001 • PET 2001

ANNOUNCING

NORTH-STAR ™

BUSINESS SOFTWARE

INVENTORY CONTROL handles 1200 items per disk. Full prompting. Can be used by untrained personel. Pays commissions & bonus based on profits, not gross sales! (Up to 5 salesmen.) Detailed reports. Includes **SUP-R-SORT.** 13 programs in reliable use over one yr. Requires 32 K, 1 drive, any printer, and any TV terminal.

INVENTROL w manual & listings	$ 89.00
MANUAL (refundable w. purchase)	$ 29.00
SUP-R-SORTs 100 items per min!	$ 39.00
ACCOUNTS RECEIVABLE	$ 89.00
ACCOUNTS PAYABLE w. manuals.	$ 89.00

Write for details. MC & Visa OK. Dealer plan.

YOUNG'S

P.O. BOX 5207 Stn. B, Victoria B.C. CANADA

ATTENTION
NORTH STAR USERS

TFS—

Text formatting system. At last a powerful, full-featured text processor for the Northstar Disc System!! "TFS" runs with virtually any system configuration and any terminal. (Note: Uses RAM from 0 to 2000H). Justifies left and right margins, page and chapter headings, page numbering, multi-copy, move text, search text, substitute, merge files, and append files from disc, and much much more! "TFS" is of great use by anyone that wants perfect text and typing with the minimum effort! Includes a 35-page user's manual complete with examples, plus a handy quick reference guide to the "TFS" commands.
= = = > > $75.00

- **TINY PASCAL:** The Chung/Yuen Tiny Pascal includes object and source! Needs only 24K to run! . . . .
  "This is a great package!" = = = > > $40.00
- **ARIAN:** 8080/Z-80 Operating System, Assembler, Text Editor Debugger, Dynamic File Capabilities, Save & Load Source/OBJ Files to Disk, Resident and Custom Commands. Truly a powerful, yet easy to use Operating System
  = = = · · $50.00

Plus much, much more, write for catalog.

(Illinois residents add 5% sales tax)

217-344-7596

SUPERSOFT ✓S61

P.O. Box 1628, Champaign, IL 61820

Limited by BASIC's logical structure?

Now available a Structured BASIC Pre-Compiler

SBASIC

SBASIC supports 5 types of loop structures, 3 logical test structures, a case structure, and procedures. Written in Microsoft Disk BASIC, it produces Microsoft Disk BASIC compatible code. CP/M compatible source micro disk $50, source listing $35 manual $10.

ULTIMATE COMPUTER SYSTEMS

313 Meadow Lane
Hastings, Michigan 49058
(313) 945-5334

(Dealer inquiries invited) ✓U12

Professional

Real Estate Programs

available now on cassette

for Apple & TRS-80 II

- **Property Managment System**
  (Two 8K Programs)
  1.) **Expense Analysis**
  2.) **Income Info & Tracking**

Introductory Offer $50
Price After July 15, $75

- **Home Purchase Analysis**
- **Income Property Cashflow/Leverage**
- **Construction Cost/Profit**
- **Tax Free Exchange Model**

Separately $20 All $65

✓R33

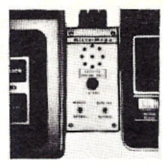
Software Realty Company

Call (213) 372-9419 for immediate C.O.D.
Attractive Dealer Pricing

2045 Manhattan Avenue, Hermosa Beach, California 90254

TRS-80
Micro-Mega CASSETTE CONTROL UNIT

● Speed up your cassette tape handling ● Pinpoint program locations on tape with an audible auxiliary monitor ● Get protection from recording and playback glitches resulting from ground loops ● Eliminate the tedious plugging and unplugging of recorder cables.



The Micro-Mega Cassette Control Unit does all this and more. You get instant manual control of the recorder at the flick of a switch. Want to find the beginning or end of a program? Flick another switch, and you'll hear it. All cables remain plugged in all the time.

The Micro-Mega Cassette Control Unit does a lot to clean up and neaten up your whole TRS-80 installation, too. As shown at the left, it's in a 2½" x 5" box which snuggles between the keyboard and your recorder. There is no need to move the recorder, and there no longer are any 90 degree cable bends whipping around. It's fast, neat and convenient—and it's a bargain!

CASSETTE CONTROL UNIT— $37.50

Micro-Mega ✓M54

P.O. BOX 6265 ● ARLINGTON VA 22206

(Virginia residents add 4% sales tax)

BASIC SUBROUTINE LIBRARY Volume 1

- ARRAYS - minimum, maximum, sums, searching, sorting.
- MATRIX ALGEBRA - addition, subtraction, multiplication, transposition, determinant, inversion.
- SIMULTANEOUS EQUATIONS - non-symmetrical, symmetrical, banded, three-diagonal.

Any routine you'll ever need, involving arrays, matrix algebra or simultaneous equations, can be found in this comprehensive collection of subroutines, compatible with any version of BASIC having subscripted variables.

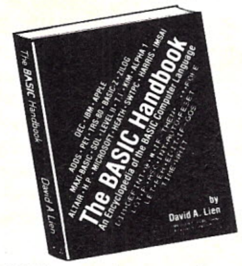
Numerous examples show how to incorporate those subroutines into your programs or how to use them as stand alone programs.

Order now and save days of work.

$18.00 U.S. plus $2.00 for handling and postage.

VM Professional Application
✓V22 Software, Inc.
2703 Bainbridge Avenue
BURNABY, B.C. V5A 2S7

The Missing Link



. . . that helps convert programs from magazines and other computers to RUN on yours. Written by the author of the widely acclaimed TRS-80 Users/Learners Manual, it explains in detail over 250 BASIC words, statements and symbols used in over 50 micros, minis and mainframes. Already in its second big printing. 360 pages.

At your dealer, or send $14.95 + $1.35 P & H (CA add 6%) to:

CompuSoft™ Publishing

✓C109 8643 Navajo Road - K7
San Diego, CA 92119

MORE INNOVATIONS!

FROM

P.S. SOFTWARE HOUSE

FORMERLY PETSHACK

PET™ INTERFACES

NEW!

PET to CENTRONICS INTERFACE	$98.00
PET to PARALLEL INTERFACE with 5V .8A power supply	$74.95
PET to 2nd CASSETTE INTERFACE	$49.95

PET™ SCHEMATICS

FOR ONLY $24.95 YOU GET:
24" X 30" schematic of the CPU board, plus oversized schematics of the Video Monitor and Tape Recorder, plus complete Parts layout - all accurately and painstakingly drawn to the minutest detail.

PET™ ROM ROUTINES

FOR ONLY $19.95 YOU GET:
Complete Disassembly listings of all 7 ROMS, plus identified subroutine entry points; Video Monitor, Keyboard routine, Tape Record and Playback routine, Real Time Clock, etc. To entice you we are also including our own Machine Language Monitor program for your PET using the keyboard and video display.
You can have the Monitor program on cassette for only $9.95 extra.

SOFTWARE:

6502 DISASSEMBLER	$12.95
MAILING LIST - For personal or business applications.	$9.95
MACHINE LANGUAGE MONITOR - Write Machine Code. Save on tape	$9.95
BUDGET - NEW - Keep track of Bills and Checks. Update as needed	$14.95
STARTREK - All-time favorite written for the PET's special Graphics	$7.95

Send for our free SOFTWARE BROCHURE. Dealer inquiries welcome.

P.S. SOFTWARE HOUSE ✓P37

P.O. Box 966 Mishawaka, IN 46544

Tel: (219) 255-3408

PET is a trademark of Commodore Business Machines

✓ I36

real-time clock

With interrupt interval-timer for S100 system. Infosystem Design's CCIT-100 simplifies your real-time programming applications.

APPLICATIONS

- Security and energy management
- Scheduling and reservation systems
- Laboratory and industrial process control
- Computer time-sharing and simulation
- Resource-use accounting
- Data logging

ADVANTAGES

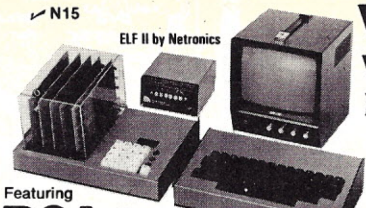
Completely assembled and tested • All software drivers included to set and read the CCIT-100 • Time and interrupt range from .01 seconds to 273 years • CPU independent • On-board crystal time base • Requires only two I/O ports

$199

Kentucky residents add 5% Sales Tax • Price includes shipping via UPS within USA • VISA • Master Charge • Money Order • COD

infosystem design inc.
809 East Broadway
Louisville, Kentucky 40204
(502) 589-2113

✓ N15



Write and run programs–the very first night–even if you've never used a computer before!

You're up and running with video graphics for just $99.95 — then use low cost add-ons to create your own personal system that rivals home computers sold for 5-times ELF II's low price!

Featuring
RCA 1802 COSMAC CPU

Own a powerful home computer system, **starting for just $99.95**–a price that gets you up and running the very first night...with your own TV for a video display. $99.95 ELF II includes RCA 1802 8-bit microprocessor addressable to 64k bytes with DMA, interrupt, 16 registers, ALU, 256 byte RAM, full hex keyboard, two digit hex output display, stable crystal clock for timing purposes, RCA 1861 video IC to display your programs on any video monitor or TV screen *and 5-slot plug-in expansion bus (less connectors) to expand ELF II into a giant!*

ELF II Explodes Into A Giant!

Master ELF II's $99.95 capabilities, then expand with GIANT BOARD ... KLUGE BOARD...4k RAM BOARDS...TINY BASIC...ASCII KEYBOARD... LIGHT PEN...ELF-BUG MONITOR...COLOR GRAPHICS & MUSIC SYSTEM... TEXT EDITOR...ASSEMBLER...DISASSEMBLER...VIDEO DISPLAY BOARD ...*and–*

More Breakthroughs Coming Soon!

Soon to be introduced: ELF II special application kits that give you the hardware and software you need to use ELF II for specialized purposes such as a telephone dialer...industrial controller...home photography...security system...police alert...motor controller...station output monitor on a conveyor belt assembly line...and some new, super-fantastic games!

Also coming soon: PROM Programmer...A-D, D-A Converter...Controller Board...and more! Unlike some heavily advertised hobby computers, ELF II doesn't limit you to pre-recorded programs. With ELF II you learn computing from the ground up...from machine language to assembly language to BASIC ...in quick, clear and easy steps. ELF II is a powerful computing tool, but one that you can master with the same ease you once mastered a slide rule or pocket calculator.

Master This Computer In A Flash!

Regardless of how minimal your computer background is now, you can learn to program an ELF II in almost no time at all. Our *Short Course On Microprocessor & Computer Programming*–written in non-technical language–guides you through each of the RCA COSMAC 1802's capabilities, so you'll understand *everything* ELF II can do...*and how to get ELF II to do it!* Don't worry if you've been stumped by computer books before. The *Short Course* represents a major advance in literary clarity in the computer field. You don't have to be a computer engineer in order to understand it. Keyed to ELF II, it's loaded with "hands on" illustrations. When you're finished with the *Short Course*, neither ELF II nor the RCA 1802 will hold any mysteries for you.

In fact, not only will you now be able to use a personal computer creatively, you'll also be able to read magazines such as BYTE...INTERFACE AGE...POPULAR ELECTRONICS and PERSONAL COMPUTING and fully *understand* the articles. And, you'll understand how to expand ELF II to give you the exact capabilities you need!

If you work with large computers, ELF II and the *Short Course* will help you understand what they're doing.

Get Started For Just $99.95, Complete!

$99.95 ELF II includes all the hardware and software you need to start writing and running programs at home, displaying video graphics on your TV screen and designing circuits using a microprocessor-the very first night–even if you've never used a computer before.

ELF II connects directly to the video input of your TV set, without any additional hardware, Or, with an $8.95 RF modulator (see coupon below), you can connect ELF II to your TV's antenna terminals instead.

ELF II has been designed to play all the video games you want, including a fascinating new target/missile gun game that was developed specifically for ELF II. But games are only the icing on the cake. The real value of ELF II is that it gives you a chance to write machine language programs–and machine language is the fundamental language of all computers. Of course, machine language is only a starting point. You can also program ELF II with assembly language and tiny BASIC. But ELF II's machine language capability gives you a chance to develop a working knowledge of computers that you can't get from running only pre-recorded tape cassettes.

ELF II Gives You The Power To Make Things Happen!

Expanded, ELF II can give you more power to make things happen in the real world than heavily advertised home computers that sell for a lot more money. Thanks to an ongoing committment to develop the RCA 1802 for home computer use, the ELF II products–being introduced by Netronics–keep you right on the outer fringe of today's small computer technology. It's a perfect computer for engineering, business, industrial, scientific and personal applications.

Plug in the **GIANT BOARD** to record and play back programs, edit and debug programs, communicate with remote devices and make things happen in the outside world. Add **Kluge** (prototyping) **Board** and you can use ELF II to solve special problems such as operating a complex alarm system or controlling a printing press. Add **4k RAM Boards** to write longer programs, store more information and solve more sophisticated problems.

ELF II add-ons already include the **ELF II Light Pen** and the amazing **ELF-BUG Monitor**–two extremely recent breakthroughs that have not yet been duplicated by any other manufacturer.

The ELF-BUG Monitor lets you debug programs with lightening speed because the key to debugging is to know what's inside the registers of the microprocessor. And, with the ELF-BUG Monitor, instead of single stepping through your programs, you can now display the entire contents of the registers on your TV screen. You find out immediately what's going on and can make any necessary changes.

The incredible ELF II Light Pen lets you write or draw anything you want on a TV screen with just a wave of the "magic wand." Netronics has also introduced the ELF II Color Graphics & Music System–more breakthroughs that ELF II owners were the first to enjoy!

ELF II Tiny BASIC

Ultimately, ELF II understands only machine language–the fundamental coding required by all computers. But, to simplify your relationship with ELF II, we've introduced an **ELF II Tiny BASIC** that makes communicating with ELF II a breeze.

Tiny BASIC saves you the time of having to code your individual instructions in machine language for ELF II. Instead, you simply type instructions on a keyboard –PRINT, RUN, LOAD, ETC. Your Tiny BASIC program automatically translates them into machine language for ELF II. Then it translates ELF II's output back into simple words and symbols for you.

Now Available! Text Editor, Assembler, Disassembler And A New Video Display Board!

The Text Editor gives you word processing ability and the ability to edit programs or text while it is displayed on your video monitor. Lines and characters may be quickly inserted, deleted or changed. Add a printer and ELF II can type letters for you–error free–plus print names and addresses from your mailing list!

ELF II's **Assembler** translates assembly language programs into hexidecimal machine code for ELF II use. The Assembler features mnemonic abbreviations rather than numerics so that the instructions on your programs are easier to read–this is a big help in catching errors.

ELF II's **Disassembler** takes machine code programs and produces assembly language source listings. This helps you understand the programs you are working with...and improve them when required.

The new **ELF II Video Display Board** lets you generate a sharp, professional 32 or 64 character by 16 line upper and lower case display on your TV screen or video monitor–dramatically improving your unexpanded $99.95 ELF II. When you get into longer programs, the Video Display Board is a real blessing!

Ask Not What Your Computer Can Do... But WHAT CAN IT DO FOR YOU?

Don't be trapped into buying an expensive dinosaur, simply because you can afford it. ELF II is more advanced and more fun to use than big name computers that cost a lot more money. With ELF II you learn to write and run your own programs. You're not just a keypunch operator. No matter what your interests are, ELF II is the fastest way to get into computers. ***Order from the coupon below!***

Netronics R&D Ltd., Dept KB-7
333 Litchfield Road, New Milford, CT 06776

PHONE ORDERS ACCEPTED!
Call (203) 354-9375

Yes! I want my own computer! Please rush me—

☐ RCA COSMAC ELF II kit at $99.95 plus $3 postage and handling (requires 6.3 to 8 volt AC power supply).

☐ Power Supply (required). $4.95 postpaid.

☐ RCA 1802 User's Manual. $5 postpaid.

☐ Tom Pittman's **Short Course On Microprocessor & Computer Programming** teaches you just about everything there is to know about ELF II or any RCA 1802 computer. Written in non-technical language, it's a learning breakthrough for engineers and laymen alike. $5 postpaid.

☐ Deluxe Metal Cabinet with plexiglas dust cover for ELF II, $29.95 plus $2.50 p&h.

☐ I am also enclosing payment (including postage & handling) for the items checked below!

☐ **I want my ELF II wired and tested with power supply, RCA 1802 User's Manual and Short Course—all for just $149.95 plus $3 p&h.**

Total Enclosed $ ______
(Conn. res. add tax)

CHARGE IT! Exp. Date ______
☐ Visa ☐ Master Charge
(Bank # ____)

Account # ______

ALSO AVAILABLE FOR ELF II

☐ **GIANT BOARD™** kit with cassette I/O, RS 232-C/TTY I/O, 8-bit P I/O, decoders for 14 separate I/O instructions and a system monitor/editor. **$39.95** plus $2 p&h

☐ **Kluge** (Prototype) **Board** accepts up to 36 IC's. **$17.00** plus $1 p&h.

☐ **4k Static RAM kit.** Addressable to any 4k page to 64k. **$89.95** plus $3 p&h.

☐ Gold plated **86-pin connectors** (one required for each plug-in board). **$5.70** ea., postpaid.

☐ **Expansion Power Supply** (required when adding 4k RAM). **$34.95** plus $2 p&h.

☐ **Professional ASCII Keyboard** kit with 128 ASCII upper/lower case set, 96 printable characters, onboard regulator, parity, logic selection and choice of 4 handshaking signals to mate with almost any computer. **$64.95** plus $2 p&h.

☐ **Deluxe metal cabinet for ASCII Keyboard, $19.95** plus $2.50 p&h.

☐ **Video Display Board** kit lets you generate a sharp, professional 32 or 64 character by 16 line upper and lower case display on your tv screen or video monitor—dramatically improving your unexpanded $99.95 ELF II. (Fits inside ASCII Keyboard cabinet.) **$89.95** plus $2 p&h.

☐ **ELF II Tiny BASIC** on cassette tape. Commands include SAVE, LOAD, ±, ×, ÷, (), 26 variables A-Z, LET, IF/THEN, INPUT, PRINT, GO TO, GO SUB, RETURN, END, REM, CLEAR, LIST, RUN, PLOT, PEEK, POKE. Comes fully documented and includes alphanumeric generator required to display alphanumeric characters directly on your tv screen without additional hardware. Also plays tick-tack-toe plus a drawing game that uses ELF II's hex keyboard as a joystick. 4k memory required. **$14.95** postpaid.

☐ Tom Pittman's **Short Course on Tiny Basic** for ELF II, **$5** postpaid.

☐ **ELF-BUG™ Deluxe System Monitor** on cassette tape. Allows displaying the contents of all registers on your tv at any point in your program. Also displays 24 bytes of memory with full addresses, blinking cursor and auto scrolling. A must for the serious programmer! **$14.95** postpaid.

☐ **Text Editor** on cassette tape gives you the ability to insert, delete or edit lines and words from your programs while they are displayed on your video monitor. (Add printer and you can use ELF II to type error-free letters plus insert names and addresses from your mailing list.) **$19.95** postpaid.

☐ **Assembler** on cassette tape translates assembly language programs into hexidecimal machine code for ELF II use. Mnemonic abbreviations for instructions (rather than numerics) make programs easier to read and help prevent errors. **$19.95** postpaid.

☐ **Disassembler** on cassette tape takes machine code programs and produces assembly language source listings to help you understand and improve your programs. **$19.95** on cassette tape.

SAVE $9.90—Text Editor, Assembler & Disassembler purchased together, only **$49.95!** (Require Video Display Board plus 4k memory.)

☐ **ELF II Light Pen**, assembled & tested. **$7.95** plus $1 p&h.

☐ **ELF II Color Graphics & Music System Board** kit. **$49.95** plus $2 p&h.

☐ ELF II connects directly to the video input of your tv set without additional hardware. To connect ELF II to your antenna terminals instead, order **RF Modulator**, **$8.95** postpaid.

Coming Soon: A-D, D-A Converter, Controller Board and more!

Print Name ______

Address ______

City ______

State ______ Zip ______

DEALER INQUIRIES INVITED

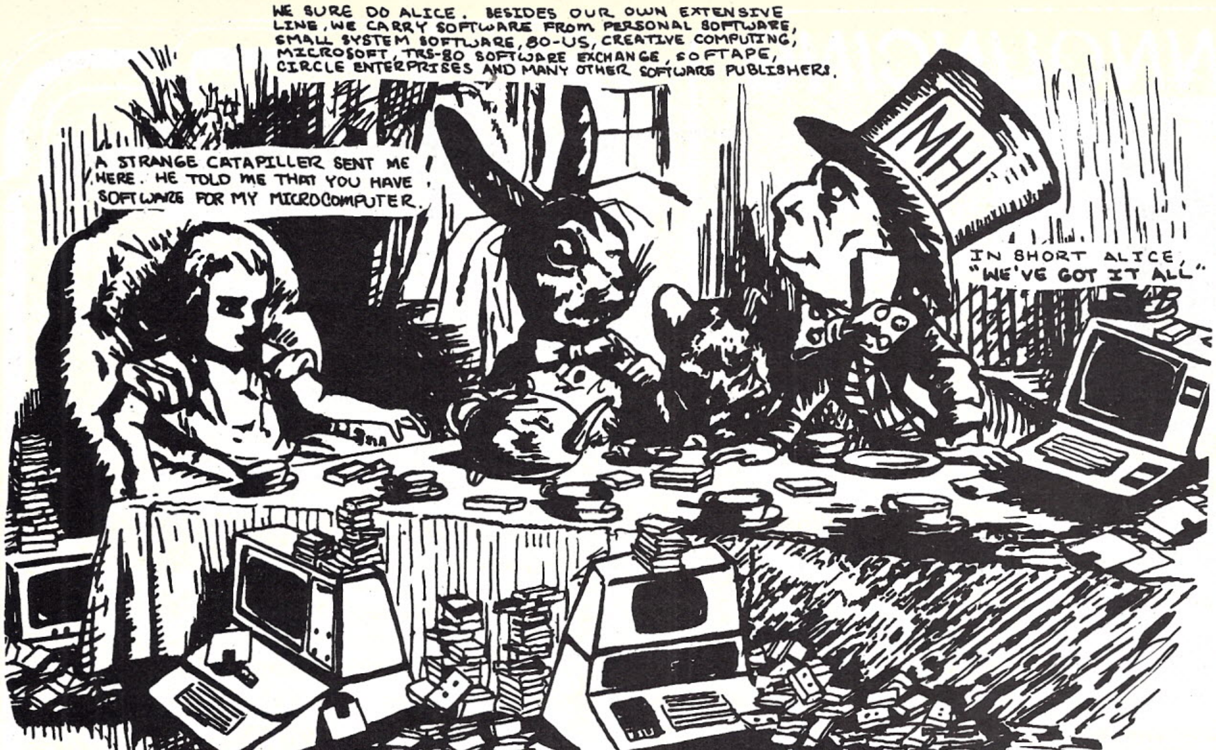
Reader Service—see page 179

Reader Service—see page 179

✓ Reader Service—see page 179

✓ Reader Service—see page 179

MAD HATTER SOFTWARE



PROGRAM	DESCRIPTION	TRS-80 L1/2	TRS-80 L2	APPLE	PET	PRICE
GALACTIC BLOCKADE RUNNER	• AN EXCITING SPACE WAR GAME WITH GRAPHICS	•		•	•	$ 9.95
SCI-FI GAME SAMPLER	• 3 GAMES—LUNAR LANDER—STAR MONSTER—SPACE BATTLE	•		•	•	$ 7.95
R/T LUNAR LANDER	• A REAL TIME LUNAR LANDER WITH GRAPHICS		•			$ 7.95
MICRO-TEXT EDITOR	• FORMAT TEXT—SAVE & LOAD TO TAPE—OUTPUT TO PRINTER		•			$ 9.95
OTHELLO III	• A STRATEGY BOARD GAME—PLAY AGAINST COMPUTER OR OTHERS	•		•	•	$ 7.95
AIR RAID	• A REAL TIME, ARCADE TYPE SHOOTING GAME IN MACH. LANG.	•				$ 14.95
MICRO-CHESS	• PLAY CHESS WITH YOUR COMPUTER—VARIOUS LEVELS OF DIFF.	•		•	•	$ 19.95
BRIDGE CHALLENGER	• DON'T WAIT FOR OTHERS TO PLAY—YOUR COMPUTER'S READY	•		•	•	$ 14.95
APPLE 21	• BLACKJACK WITH HIRES GRAPHICS			•		$ 9.95
STAR WARS/SPACE MAZE	• SCI-FI GAMES FOR THE APPLE			•		$ 12.95
RENUMBER	• RENUMBER YOUR BASIC PROGRAMS—RENUMBERS EVERYTHING	•		•		$ 14.95
DISK RENUMBER	• SAME AS ABOVE, BUT ON DISK	•				$ 19.95
PILOT 2.0	• THE EDUCATIONAL LANGUAGE, IN MACH. LANG.—INC. EDITOR		•			$ 14.95
PILOT 3.0	• THE DISK VERSION OF THE ABOVE		•			$ 24.95
APPLE TALKER	• YOUR APPLE SPEAKS! NO NEW HARDWARE REQUIRED			•		$ 15.95
APPLE LIS'NER	• SPEECH RECOGNITION THE EASY WAY—GREAT WITH THE TALKER			•		$ 19.95
TIC-TAC-TALKER	• TIC-TAC-TOE USING SPEECH SYNTHESIS AND RECOGNITION			•		$ 19.95
FORTRAN	• FOR THE TRS-80—SEE MICROSOFT'S ADS		•			$325.00
SYSCOP	• MAKE BACKUP TRS-80 SYSTEM TAPES THE EASY WAY		•			$ 9.95
ANDROID NIM-2	• GAME OF NIM WITH ANIMATED ROBOTS AND SOUND		•			$ 14.95
SNAKE EGG	• A BETTING GAME WITH ANIMATED SNAKES AND SOUND		•			$ 14.95
LIFE 2	• 100 GEN. PER MIN. LIFE & BATTLE OF LIFE W/ANIMATION & SOUND		•			$ 14.95
DCV-1	• PUT SYSTEM TAPES ON DISK EVEN IF IN SAME MEM AS DOS		•			$ 9.95
MUSIC MASTER	• ENTER SHEET MUSIC—THE TRS-80 THEN COMPILES & PLAYS IT		•			$ 14.95
DISK MUSIC MASTER	• SAME AS ABOVE BUT ON DISK W/MANY SELECTIONS		•			$ 24.95
TRS-80 CP/M	• OPENS UP THE WHOLE WORLD OF CP/M SOFTWARE TO THE TRS-80		•			$150.00
PET ASTROLOGY	• DOES ALL THE COMPUTATIONS FOR YOU				•	$ 14.95
PERSONAL FINANCE PKG.	• KEEPS TRACK OF CHECKS, BUDGET, COMPUTES INTEREST	•		•	•	$ 9.95
TRS-80 DISK LIB. "A"	• 9 PROGRAMS ON DISK—INC. BLOC. RUN, OTHELLO ETC.		•			$ 39.95
BUSINESS AND APPLICATION SOFTWARE—CALL OR WRITE FOR DETAILS						
16K MEMORY UPGRADE FOR TRS-80, APPLE & SORCERER COMPUTERS						$ 94.95
C-10 HIGH QUALITY DATA CASSETTES W/SCREW HOUSINGS						10/$ 34.95
VERBATUM 5-1/2 INCH BLANK DISKETTES						10/$ 10.00

10% OFF IF YOU ORDER 3 SOFTWARE PACKAGES OR MORE **SEND FOR FREE CATALOG—GIVE TYPE OF COMPUTER**

AVAILABLE FROM THESE FINE MICRO COMPUTER DEALERS

J&J ELECTRONICS LTD
28 COLLINGTON AVE.
BEXHILL-ON-SEA, E. SUSSEX, ENG.

HOBBY WORLD ELECTRONICS
19355 BUSINESS CENTER DR. #6
NORTHRIDGE CA 91324

CAPITOL COMPUTER SYSTEMS
3396 EL CAMINO AVE.
SACRAMENTO CA 95821

COMPUTER CABLEVISION, INC.
2617 42ND ST. NW #2
WASHINGTON DC 20007

COMPUTER VILLAGE
931 SW 87TH AVE.
MIAMI FL 33174

AUGUST AUTOMATION
28 MILK ST.
WESTBORO MA 01581

TRS-80 SOFTWARE EXCHANGE
17 BRIARCLIFF DR.
MILFORD NH 03055

KENNEDY SYSTEMS
74 BROAD ST.
LYNDONVILLE VT 05851

L. C. SALES
100 HINCHEY AVE. #705
OTTAWA, ONT., CAN. K1Y4L9

THE CPU SHOP
39 PLEASANT ST.
CHARLESTOWN MA 02129

OP AMP TECH BOOKS
1033 N. SYCAMORE AVE.
LOS ANGLES CA 90038

ADVANCED COMPUTER PRODUCTS
1310B E. EDINGER
SANTA ANA CA 92705

TO ORDER BY PHONE OR FOR DEALER INFO—CALL—(617) 682-8131
ADD 75c SHIPPING & HANDLING • MASS. RESIDENTS ADD 5% SALES TAX
MAD HATTER SOFTWARE • 900 K SALEM RD • DRACUT, MA 01826 ✓M77

ANNOUNCING

New from DEC

LA34 DECwriter IV

$1,199.00

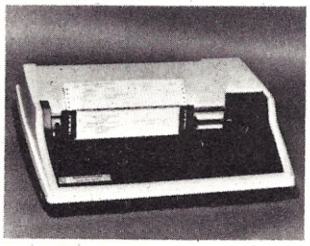
- 110 or 300 baud, RS 232C serial ASCII
- Friction feed, paper width to 15"
- Upper/lower case, 9 x 7 dot matrix
- 10, 12, 13.2, 16.5 characters /inch
- 2, 3, 4, 6, 8, or 12 lines/inch
- 22"W x 7"H x 15½"D, 25 lbs.

Teletype 43 $ 999.00

- RS 232C, 110 or 300 baud
- Upper/lower case full ASCII
- Pin feed, 12" x 8½" paper

T.I. 810 printer $1,695.00

- 150 characters per second
- RS 232C serial interface
- Adjustable forms tractor
- Upper/lower case option $90.00



New from DIABLO

DIABLO 1640 $2,690.00

Receive-only $ 2,331.00

High-quality daisywheel printing at 45 cps.

DIABLO 1650 $ 2,779.00

Receive-only $ 2,419.00

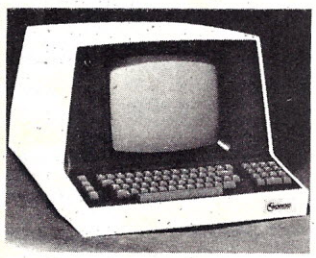
Metal daisywheel printing at 40 cps.

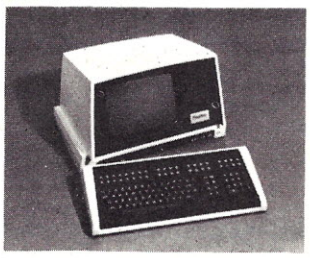
SOROC IQ 120 $ 795.00

- RS 232C, upper/lower case full ASCII
- Numeric keypad, protected fields
- Cursor keys plus addressable cursor
- Auxillary extension port

SOROC IQ 140 $1,250.00

- RS 232C and 20mA current loop
- Extensive editing features
- 25th line terminal status display
- 16 function keys (32 with shift)





To Order: Send certified check (personal or company checks require two weeks to clear) including handling* and 6% sales tax if delivered within California.

*Handling: Less than $2,000, add 2%; over $2,000, add 1%. Everything shipped freight collect in factory cartons with manufacturer's warranty.

Visit MICROMAIL at the National Computer Conference — Personal Computing Festival, New York City, June 5-7, Booth 128.

MICROMAIL

MICROMAIL • BOX 3297 • SANTA ANA, CA 92703

(714) 731-4338 ✓M73

Fantastic New TARBELL DISK BASIC Only $48.00

- Runs on 8080, 8085, or Z80
- Searches a file quickly for a string.
- Up to 64 files open at once.
- Random access.
- Assignment of I/O.
- Alphanumeric line labels are allowed.
- Read and write string or numeric data.
- Unlimited length of variable names and strings.
- Procedures with independent variables.
- Number system 10 digits BCD integer or floating point.
- Chain to another program.
- Cause programs to be appended onto programs already in memory.
- Cause interpreter to enter edit mode using 15 single character edit commands.

Tarbell BASIC occupies 24K of RAM. Tarbell BASIC on CP/M* Disk $48. Source on paper or CP/M Disk $25.

*CP/M is a Trademark/Tradename of Digital Research.

✓T11

Tarbell Electronics

950 DOVLEN PLACE, SUITE B
CARSON, CA 90746
(213) 538-4251 • (213) 538-2254

TRS - 80™

NEW DOS UPDATED TRS DOS	**$49.00**
NEW DOS† SAME WITH UTILITIES	**$99.00**
K.V.P. KEYBOARD, VIDIO, & PRINTER SUBROUTINES	**$29.00**
LEVEL IVGRAPHICS FOR WRITTING SUPER-FAST & EASY GRAPHICS AS IN PROGRAMS LISTED BELOW	**$15.00**
ANIMATED HANGMAN HE WINKS, BLINKS AND MORE	**$10.00**
TYPING 1 LEARN TO TYPE (Don't miss this one!)	**$15.00**
SPACE BATTLE ONE OF THE ALL TIME GREAT TRS-80 GAMES (cass.)	**$14.95**
SPACE BATTLE SAME, (But for Disk)	**$19.95**

(All Programs Listed Above Are L II or Disk Only)

-Dealer Inquiries Invited-

DISTRIBUTED BY:

LEVEL IV PRODUCTS, INC.

32020 CHEBOYGAN
WESTLAND, MICHIGAN 48185
(313) 326-0386

VISA · master charge

✓ Reader Service—see page 179

10% OFF 20% OFF

YOUR OWN TRS-80 SYSTEM AT TREMENDOUS SAVINGS

①TRS-80 Complete System
Includes: CPU/Keyboard, Power Supply, Video Monitor, Cassette Recorder, Manual, and Game Cassette.
②Line Printer
③Mini Disk System
④C-10 Cassettes
⑤Verbatum Diskettes

DISK DRIVES NOW IN STOCK!

ITEM		REG. PRICE	OUR PRICE
TRS-80 Complete System Level II - 4K RAM		$ 698.00	$ 628.20
TRS-80 Complete System Level II 16 K RAM		$ 988.00	$ 889.20
Expansion Interface		$ 299.00	$ 269.10
Pertec FD2000 Disk Drive		$ 495.00	$ 385.00
BASF 6106		$ 599.00	$ 495.00
Centronics 779 Printer		$1599.00	$1299.00
Centronics 101 Printer		$1595.00	$1400.00
Anadex DP-8000 Printer		$1295.00	$ 995.00
Centronics P1 Printer		$ 534.00	$ 445.00
Trendata 1000		$1495.00	$1295.00
Memory Unit (Kit) *FREE INSTALLATION		$ 199.00	$ 96.00
Verbatum Diskettes	ea.	$ 5.95	$ 4.95
	3	$ 17.89	$ 12.00
	10	$ 59.00	$ 37.00
Maxwell Diskettes	ea.	$ 10.00	$ 7.50
	3	$ 30.00	$ 21.00
	10	$ 100.00	$ 60.00
C-10 Cassettes	5	$ 4.95	$ 4.50
	25	$ 24.75	$ 18.75
C-30 Cassettes	12	$ 29.95	$ 23.95
Paper (9½" x 11" fanfold, 3500 sheets)		$ 35.00	$ 29.95

NEW & IMPORTANT SOFTWARE

Lower Case conversion - Level II - Tape – **$19.95**
- Type text on your keyboard just as you would on your typewriter
- For use with hard copy devices with a lower case option

Loan Amortization Schedule - Level II - Tape/Disk - 16K - **$19.95**
- Variable Period Length
- Double Precision
- Prints one period or entire schedule
- Final Payment Adjusted

Real Estate Package - Level II - Disk - 16K **$34.95**
- Listing & Buyer Files * Mortgage type
- Sorts by * Price ± 5% * Location
  * # of Bedrooms.

Household Inventory - Level II - Tape - 16K - **$19.95**
- For Insurance claims
- Classified by Room, Purchase date, & Price
- Add, Detete, Change Routines

Business software
for TRS—80

Business Inventory - Level II - 32K - 2 Disk Drives **$49.95**
- Random Access - Double Precision - 650 Item
- FIELDS
  Item Description, Unit cost, Selling Price, Transaction date, Catalog #, P.O. #, Qty on Hand, Reorder level, #Sold, Daily & Total Activity
- Reports Printed -
  Inventory - Activity to Date
  Re-order List - On-order List - Out of Stock List
  Dept/Sales Report - Inventory Worksheets

Comprehensive Business Data Base - Level II - 32K - 2 Disk Drives **$129.50**
Includes Inventory Above plus:
- Account Number Generation (Check digit)
- Daily Transaction & Trial Balance
- Statement of Accounts & Aging Report
- Inventory Inquiry
- Invoicing

Avail. soon for Data Base - General Ledger, P/L, Tax Reports

Payroll (115 employees) - Level II - Disk - 32K **$49.95**
- Fed-with, FICA, State, Local tax - Deducts sick pay from FICA
- Month, Qty, Year to date reports -
- Random Access - Double Precision - Can change individual fields
- Prints Checks & Stubs

VR Data corp. ✓V19 *MORE DATA PER DOLLAR*
VISA master charge
777 Henderson Boulevard N-6 Folcroft Industrial Park Folcroft, PA 19032 (215) 461-5300

Reader Service—see page 179

P.O. Box 4430M Santa Clara, CA 95054
For will call only: (408) 988-1640
2322 Walsh Ave.

Quest ELECTRONICS

✓Q3

RCA Cosmac Super Elf Computer $106.95

Compare features before you decide to buy any other computer. There is no other computer on the market today that has all the desirable benefits of the **Super Elf** for so little money. The Super Elf is a small single board computer that does many **big** things. It is an excellent computer for training and for learning programming with its machine language and yet it is easily **expanded with additional memory, Tiny Basic, ASCII Keyboards, video character generation, etc.**

The **Super Elf** includes a **ROM** monitor for program loading, editing and execution with **SINGLE STEP for program debugging** which is not included in others at the same price. With **SINGLE STEP** you can see the microprocessor chip operating with the **unique Quest** address and data bus displays **before, during** and **after** executing instructions. Also, CPU mode and instruction cycle are decoded and displayed on eight LED indicator lamps.

An RCA 1861 **video graphics** chip allows you to connect to your own TV with an inexpensive video modulator to do graphics and games. There is a **speaker system** included for writing your own music or using many music programs already written. The speaker amplifier may also be used to drive relays for control purposes.

A 24 key **HEX keyboard** includes 16 HEX keys plus **load, reset, run, wait, input, memory protect, monitor select** and **single step**. Large, on board displays provide output and optional **high and low address**. There is a 44 pin standard connector for PC cards and a 50 pin connector for the **Quest Super Expansion Board**. Power supply and sockets for **all** IC's are included in the price plus a detailed 127 pg. instruction manual which now includes over 40 pgs. of software info. including a series of lessons to help get you started and a music program and graphics target game.

Remember, other computers only offer Super Elf features at additional cost or not at all. **Compare before you buy. Super Elf Kit $106.95, High address option $8.95, Low address option $9.95. Custom Cabinet** with drilled and labelled plexiglass front panel **$24.95. NiCad Battery Memory Saver Kit $6.95.** All kits and options also come completely assembled and tested.

Questdata, a 12 page monthly software publication for 1802 computer users is available by subscription for $12.00 per year.

Attention Elf Owners
New products in hardware and software coming soon.

Tiny Basic cassette $10.00, on ROM $38.00, original Elf kit board $14.95.

Super Expansion Board with Cassette Interface $89.95

This is truly an astounding value! This board has been designed to allow **you** to decide how you want it optioned. **The Super Expansion Board comes with 4K of low power RAM** fully addressable anywhere in 64K with built-in memory protect and a **cassette interface**. Provisions have been made for all other options on the same board and it fits neatly into the hardwood cabinet alongside the **Super Elf**. The board includes slots for up to 6K of **EPROM** (2708, 2758, 2716 or TI 2716) and is **fully socketed**. EPROM can be used for tbe monitor and Tiny Basic or other purposes.

A IK Super ROM Monitor $19.95 is available as an on board option in 2708 EPROM which has been preprogrammed with a program loader/editor and error checking multi file cassette read/write software, (relocatible cassette file) another exclusive from Quest. It includes register save and readout, block move capability and video graphics driver with blinking cursor. Break points can be used with the register save feature to isolate program bugs quickly, then follow with single step. The Super Monitor is written with subroutines allowing users to take advantage of monitor functions simply by calling them up.

Improvements and revisions are easily done with the monitor. If you have the **Super Expansion Board** and **Super Monitor** the monitor is up and running at the push of a button.

Other on board options include **Parallel Input and Output Ports** with **full handshake**. They allow easy connection of an ASCII keyboard to the input port. **RS 232 and 20 ma Current Loop** for teletype or other device are on board and if you need more memory there are two **S-100** slots for static RAM or video boards. A Godbout 8K RAM board is available for $135.00. Also a 1K Super Monitor version 2 with video driver for full capability display with Tiny Basic and a video interface board. **Parallel I/O Ports $9.85, RS 232 $4.50, TTY 20 ma I/F $1.95, S-100 $4.50. A 50 pin connector set** with ribbon cable is available at $12.50 for easy connection between the **Super Elf** and the **Super Expansion Board.**

The **Power Supply Kit** for the Super Expansion Board is a 5 amp supply with multiple positive and negative voltages $29.95. Add $4.00 for shipping. Prepunched frame $5.00. Case $10.00. Add $1.50 for shipping.

Same day shipment. First line parts only. Factory tested. Guaranteed money back. Quality IC's and other components at factory prices.

INTEGRATED CIRCUITS

7400TTL	
7400N	.17
7402N	.17
7404N	.19
7409N	.23
7410N	.17
7414N	.63
7420N	.17
7422N	1.39
7430N	.20
7442N	.50
7445N	.69
7447N	.60
7448N	.69
7450N	.17
7474N	.29
7475N	.49
7485N	.88
7489N	2.00
7490N	.43
7492N	.43
7493N	.43
7495N	.69
74100N	.90
74107N	.29
74121N	.34
74123N	.59
74125N	.39
74145N	.69
74150N	.95
74151N	.69
74154N	1.00
74157N	.69
74161N	.87
74162N	.87
74163N	.87
74174N	.96
74175N	.90
74190N	1.15
74192N	.87
74193N	.85
74221N	1.55
74298N	1.65
74365N	.95
74366N	.95
74367N	.95

74LS00 TTL	
74LS00N	.25
74LS02N	.25
74LS04N	.25
74LS05N	.25
74LS08N	.25
74LS10N	.25
74LS13N	.40
74LS14N	.90
74LS20N	.25
74LS22N	.25
74LS28N	.41
74LS30N	.25
74LS33N	.39
74LS38N	.30
74LS74N	1.15
74LS75N	.47
74LS90N	.51
74LS93N	.51
74LS95N	1.89
74LS107N	.35
74LS112N	.35
74LS113N	.35
74LS132N	.72
74LS136N	.35
74LS151N	.67
74LS155N	.67
74LS157N	.67
74LS162N	.91
74LS163N	.91
74LS174N	.95
74LS190N	1.06
74LS221N	1.95
74LS258N	.67
74LS367N	1.35

LINEAR	
CA3045	.90
CA3046	.67
CA3081	1.80
CA3082	1.90
CA3089	2.95
LM301AN/AH	.35
LM305H	.87
LM307N	.35
LM308N	.89
LM309H	1.15
LM309K	1.50
LM311H/N	.90
LM317T/K	2.92
LM318	1.35
LM320K-5	1.60
LM323K-5	6.95
LM320K-12	1.60
LM320K-15	1.60
LM320T-5	1.60
LM320T-8	1.60
LM320T-12	1.50
LM320T-15	1.60
LM324N	1.15
LM339N	1.55
LM340K-5	1.35
LM340K-8	1.35
LM340K-12	1.35
LM340K-15	1.35
LM340K-24	1.35
LM340T-5	1.25
LM340T-8	1.25
LM340T-12	1.25
LM340T-15	1.25
LM340T-18	1.25
LM340T-24	1.25
LM343H	4.50
LM350	7.50
LM370	1.15
LM377	3.00
LM379M	5.00
LM380N	1.00
LM381	1.60
LM382	1.60
LM703H	.40
LM709H	.28
LM723H/N	.50
LM733N	.67
LM741CH	.35
LM741N	.25
LM747H/N	.62
LM748N	.35
LM1303N	.82
LM1304	1.10
LM1305	1.27
LM1307	2.00
LM1310	2.75
LM1458	.47
LM1800	1.75
LM1812	7.50
LM1889	3.00
LM2111	1.75
LM2902	1.50
LM3900N	.60
LM3905	1.75
LM3909N	.61
MC1458V	.50
NE550N	.65
NE555V	.43
NE556A	.79
NE565A	1.00
NE566V	1.50
NE567V	1.20
NE570B	5.00
NE571B	5.00
78L05	.60
78L08	.60
79L05	.70
78M05	.85
75108	1.75
75491CN	.50
75492CN	.55
75494CN	.89

A to D CONVERTER	
8038B	4.50
8700CJ	13.95
8701CN	15.95
8750CJ	13.95
LD130	9.95
9400CJV/F	7.40
ICL7103	9.50
ICL7107	14.25
8702	17.95

CMOS	
CD34001 Fair.	.50
CD4000	.16
CD4001	.21
CD4002	.21
CD4006	1.10
CD4007	.21
CD4008	.21
CD4009	.39
CD4010	.39
CD4011	.21
CD4012	.21
CD4013	.36
CD4014	.86
CD4015	.86
CD4016	.36
CD4017	.94
CD4018	.94
CD4019	.21
CD4020	1.02
CD4021	1.02
CD4022	.86
CD4023	.21
CD4024	.75
CD4025	.21
CD4026	1.51
CD4027	.36
CD4028	.79
CD4029	1.02
CD4030	.21
CD4035	1.02
CD4040	1.02
CD4042	.71
CD4043	.63
CD4044	.63
CD4046	1.67
CD4049	.36
CD4050	.36
CD4051	1.13
CD4060	1.42
CD4066	.71
CD4068	.40
CD4069	.40
CD4070	.40
CD4071	.21
CD4072	.21
CD4073	.21
CD4075	.21
CD4076	1.75
CD4078	.40
CD4081	.21
CD4082	.21
CD4116	.47
CD4490	5.50
CD4507	1.00
CD4508	4.25
CD4510	1.02
CD4511	.94
CD4515	2.52
CD4516	1.10
CD4518	1.02
CD4520	1.02
CD4527	1.51
CD4528	.79
CD4553	3.50
CD4566	2.25
CD4583	4.50
CD4585	1.10
CD40192	3.00
74C00	.28
74C04	.33
74C10	.28
74C14	2.10
74C20	.28
74C30	.28
74C48	1.95
74C74	.75
74C76	1.40
74C90	1.15
74C93	1.40
74C154	3.00
74C160	1.44
74C175	1.35
74C192	1.65
74C221	2.00
74C905	3.00
74C906	.75
74C914	1.95
74C922	5.50
74C923	5.50
74C925	6.95
74C926	6.95
74C927	6.95

INTERFACE	
8095	.95
8096	.95
8097	.95
8098	.95
8T09	1.25
8T10	4.50
8T13	3.00
8T20	5.50
8T23	3.10
8T24	3.50
8T25	3.20
8T26	1.69
8T28	2.75
8T97	1.69
8T98	1.69

MOS/MEMORY RAM	
2101-1	3.95
2102-1	.95
2102AL-4	1.60
21L02-1	1.18
21F02	1.25
2104A-4	4.95
2107B	4.95
2111-1	3.75
2112-2	3.95
2114L-3	6.75
4116	10.95
2513B	6.30
MM5262	.40
MM5280	3.00
MM5320	9.95
MM5330	5.94
PD411D-3	4.00
PD411D-4	5.00
P5101L	13.95
4200A	9.95
82S25	2.90
91L02A	1.50
HD0165-5	6.95
MM57100	4.50
GIAY38500-1	9.95
MCM6571A	9.95
9368	3.50
410D	10.00
416	16.00

CLOCKS	
MM5309	3.00
MM5311	3.60
MM5312	4.80
MM5313	3.60
MM5314	3.90
MM5315	4.00
MM5316	5.00
MM5318	3.60
MM5369	2.10
MM5841	14.45
MM5865	7.95
CT7001	5.80
CT7002	10.95
CT7010	8.95
CT7015	8.95
MM5375AA/N	3.90
MM5375AB/N	4.90
7205	16.50
7207	7.50
7208	15.95
7209	4.95
DS0026CN	3.75
DS0056CN	3.75
MM53104	2.50

MICROPROCESSOR	
6800	17.50
6802	24.95
8080A with data	8.95
8085	27.00
Z80A	19.75
8212	2.90
8214	8.00
8216	2.90
8224	2.90
8228	5.35
8251	8.50
8253	10.00
8255	9.25
8257	19.50
8259	19.50
1802CP plas.	13.95
1802DP plas.	17.95
1861P	11.50
CDP1802CD	19.95
CDP1802D	25.00
CDP1861	12.95
6820	9.95
6850	12.95
6502	12.50

IC SOCKETS
Solder Tin Low Profile

PIN	1 UP	PIN	1UP
8	.15	22	.30
14	.14	24	.35
16	.16	28	.42
18	.27	36	.58
20	.29	40	.57

3 level wire wrap gold.
14 pin .25
2 level 14 pin ww .20

WIRE WRAP LEVEL 3

PIN		PIN	
14	.25	24	.86
16	.33	28	1.00
18	.57	40	1.23

UART/FIFO	
AY5-1013	5.50
AY5-1014	7.50
3341	6.95

PROM	
1702A	3.95
N82S23	2.95
N82S123	3.50
N82S126	3.75
N82S129	3.75
N82S131	3.75
N82S136	8.75
N82S137	8.75
2708	10.50
DM8577	2.90
8223	2.90
2716T1	29.50
2716 Intel	48.00

CRYSTALS	
1 MHz	4.50
2 MHz	4.50
4 MHz	4.25
5 MHz	4.25
10 MHz	4.25
18 MHz	3.90
20 MHz	3.90
32 MHz	3.90
32768 MHz	4.00
1.8432 MHz	4.50
3.5795 MHz	1.20
2.0100 MHz	1.95
2.097152 MHz	4.50
2.4576 MHz	4.50
3.2768 MHz	4.50
5.0688 MHz	4.50
5.185 MHz	4.50
5.7143 MHz	4.50
6.5536 MHz	4.50
14.31818 MHz	4.25
18.432 MHz	4.50
22.1184 MHz	4.50

CONNECTORS	
44 pin edge	2.75
100 pin edge	4.50
100 pin edge WW	4.75

KEYBOARD ENCODERS	
AY5-2376	$12.50
AY5-3600	13.50
74C922	5.50
74C923	5.50
HD0165-5	6.95

IC Test Clips

	1	10
Red	.55	.47
Black	.55	.47

Keyer 8043 14.50
comp. w/spec./sock.

TRANSISTORS	
2N1893	.40
2N2222A	.18
2N2369	.30
2N2904A	.20
2N2907A	.25
2N3053	.40
2N3638	.25
2N3643	.25
2N3904	.18
2N3906	.18
2N3055	.69
2N4400	.25
2N4401	.75
2N4402	.20
TIP31	.60
TIP33A	1.00

D Connectors RS232
25 Pin Subminiatures

DB25P	2.95
DB25S	3.95
Cover	1.50
RS232 Complete Set	6.50
DE9P	1.50
DE9S	1.95
DA15P	2.10
DA15S	3.10

CLOCK MODULES Complete alarm clocks ready to hook up with transformer and switches. Very compact with .50" and .84" digits.

MA1002A, C or E .50"	8.95
102P3 Transformer	2.25
MA1010A, C or E .84"	11.95
102P2 Transformer	2.25
Special transformer and six switches when purchased w/module	2.95
MA1003 car module .3" green fluor. display	15.95

RESISTORS ¼ watt 5%
10 per type .03   1000 per type .012
25 per type .025   350 piece pack
100 per type .015   5 per type 6.75

KEYBOARDS
56 key ASCII keyboard kit $67.50
Fully assembled 77.50
53 key ASCII keyboard kit 60.00
Fully assembled 70.00 Enclosure 14.95

LEDS
Red T018 .15
Green, Yellow T018 .20
Jumbo Red .20
Green, Orange, Yellow Jumbo .25
Cliplite LED Mounting Clips 8/$1.25
(specify red, amber, green, yellow, clear)

CONTINENTAL SPECIALTIES in stock
Complete line of breadboard test equip.
MAX-100 8 digit Freq. Ctr. $128.95

OK WIRE WRAP TOOLS in stock
Portable Multimeter $18.00

DIGITAL THERMOMETER $48.50
Batt. oper. General purpose or medical 32 – 230 F. Disposable probe cover ·.2 accuracy. Comp. Assy. in compact case. Switches from F to C.

COMPUTER BOARD KITS
8K RAM Board Kit $135.00
4K EPROM Kit 114.95
I/O Board Kit 44.50
Extender Board w/connector 12.50
16K EPROM board kit w/o PROMS 74.50
North Star Floppy Disk Kit $665.00
Additional Drive Kit 415.00

SPECIAL PRODUCTS
MM5865 Stopwatch Timer 9.00
PC board 7.50
Switches Mom. Pushbutton .27
3 pos. slide .25
Encoder HD0165-5 6.95
3 Digit Universal Counter Board Kit
Operates 5–18 Volt DC to 5 MHz typ. .125" LED display **10.50**
Voice actuated switch .50
Paratronics 100A Logic Analyzer Kit $235.00
Model 10 Trigger Expander Kit $229.00
Model 150 Bus Grabber Kit $369.00
Sinclair 3½ Digit Multimeter $59.95
Clock Calendar Kit $23.95

TRANSFORMERS
6V 300 ma 3.25
12 Volt 300 ma transformer 1.25
12.6V CT 600 ma 3.75
12V 250 ma wall plug 2.95
12V CT 250 ma wall plug 3.50
24V CT 400 ma 3.95
10V 1.2 amp wall plug 4.85
12V 6 amp 12.95

DISPLAY LEDS

MAN1	CA	.270	2.90
MAN3	CC	.125	.39
MAN72/74	CA/CA	.300	1.00
DL704	CC	.300	1.25
DL707/DL707R	CA	.300	1.00
DL727/728	CA/CC	.500	1.90
DL747/750	CA/CC	.600	1.95
DL750	CC	.600	1.95
FND359	CC	.357	.70
FND500/507	CC/CA	.500	1.35
FND503/510	CC/CA	.500	.90
FND800/807	CC/CA	.800	2.20
3 digit Bubble			.60
4 digit bubble			.80
DG8 Fluorescent			1.75
DG10 Fluorescent			1.75
5 digit 14 pin display			1.00
NSN69 9 digit display			.60
7520 Clairex photocells			.39
TIL311 Hex			9.50

COMPUTER GRADE CAPS

1600	mfd	200V	4.50
2000	mfd	45V	2.50
3200		50V	2.50
5500		25V	2.50
5800		40V	3.00
6100		40V	3.00
6100		50V	3.00
7700		50V	2.50
8000		55V	2.50
10000		20V	2.50
12000		30V	3.50
35000		25V	5.00
55000		25V	5.00
82000		15V	6.00
60000		40V	6.00
76000		40V	6.00
120000		10V	6.00

Auto Clock Kit $15.95
DC clock with 4-.50" displays. Uses National MA-1012 module with alarm option. Includes light dimmer, crystal timebase PC boards. Fully regulated, comp. instructs. Add $3.95 for beautiful dark gray case. Best value anywhere.

RCA Cosmac VIP Kit $229.00
Video computer with games and graphics. Fully assem. and test. **$249.00**

Not a Cheap Clock Kit $14.95
Includes everything except case. 2-PC boards. 6-.50" LED Displays. 5314 clock chip, transformer, all components and full instructions. Orange displays also avail. **Same kit w/.80" displays.** Red only. **$21.95** Case **$11.75**

60 Hz Crystal Time Base Kit $4.40
Converts digital clocks from AC line frequency to crystal time base. Outstanding accuracy. Kit includes: PC board, IC, crystal, resistors, capacitors and trimmer.

Digital Temperature Meter Kit
Indoor and outdoor. Switches back and forth. Beautiful. 50" LED readouts. Nothing like it available. Needs no additional parts for complete, full operation. Will measure −100° to +200°F, tenths of a degree, air or liquid. Very accurate. **$39.95**
Beautiful woodgrain case w/bezel **$11.75**

NiCad Battery Fixer/Charger Kit
Opens shorted cells that won't hold a charge and then charges them up, all in one kit w/full parts and instructions. **$7.25**

PROM Eraser
Will erase 25 PROMs in 15 minutes. Ultraviolet, assembled **$34.50**

Rockwell AIM 65 Computer
6502 based single board with full ASCII keyboard and 20 column thermal printer. 20 char. alphanumeric display, ROM monitor, fully expandable. $375.00. 4K version $450.00. 4K Assembler $85.00, 8K Basic Interpreter $100.00. Power supply assembled in case $60.00.

Multi-volt Computer Power Supply
8v 5 amp, ±18v .5 amp, 5v 1.5 amp, −5v .5 amp, 12v .5 amp, −12 option. ±5v, ±12v are regulated. Kit $29.95. Kit with punched frame $34.95. Woodgrain case $10.00.

Video Modulator Kit $8.95
Convert your TV set into a high quality monitor without affecting normal usage. Complete kit with full instructions.

2.5 MHz Frequency Counter Kit
Complete kit less case **$37.50**

30 MHz Frequency Counter Kit
Complete kit less case **$47.75**
Prescaler kit to 350 MHz **$19.95**

79 IC Update Master Manual $35.00
Complete IC data selector, 2500 pg. master reference guide. Over 50,000 cross references. Free update service through 1979. Domestic postage $3.50. Foreign $5.00. 1978 IC Master closeout $19.50.

Stopwatch Kit $26.95
Full six digit battery operated. 2–5 volts. 3.2768 MHz crystal accuracy. Times to 59 min., 59 sec., 99 1/100 sec. Times std., split and Taylor. 7205 chip, all components minus case. Full instructions.

Hickok 3½ Digit LCD Multimeter
Batt/AC oper. 0.1mv–1000v. 5 ranges. 0.5% accur. Resistance 6 low power ranges 0.1 ohm–20M ohm. DC curr. .01 to 100ma. Hand held, ½" LCD displays, auto zero, polarity, overrange. $74.95.

S-100 Computer Boards

8K Static RAM Kit Godbout	$135.00
16K Static RAM Kit	265.00
24K Static RAM Kit	423.00
32K Dynamic RAM Kit	310.00
64K Dynamic RAM Kit	470.00
8K/16K Eprom Kit (less PROMS)	**$89.00**
Video Interface Kit	**$139.00**
Motherboard **$39.** Extender Board	**$8.99**

TERMS: $5.00 min. order U.S. Funds. Calif residents add 6% tax. BankAmericard and Master Charge accepted. Shipping charges will be added on charge cards.

FREE: Send for your copy of our NEW 1979 QUEST CATALOG. Include 28¢ stamp.

✓ Reader Service—see page 179

✓ *Reader Service—see page 179*

Terminals and Printers

for every application . . .

MiniMicroMart supplies a complete selection of CRT terminals and printing devices for every application.

Printers are available from 30 cps to 300 lines per minute.

For word processing or report writing applications, we offer a wide variety of fast, letter-quality machines, such as the **NEC Spinwriter**, the **Diablo Hytype**, and the **Qume Sprint**. We also offer a conversion package which adapts your **Selectric Typewriter** to work as a computer printer.

Virtually every printer we sell can be used or adapted readily to any of the popular machines on the market, including the **PET, APPLE**, and **TRS-80**

— CALL FOR SPECIFIC DETAILS —

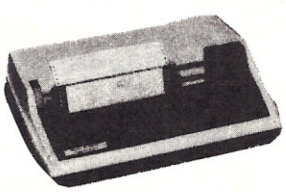
NEC **spinwriter**™

The Fantastic Letter-Quality Printer at 55 cps

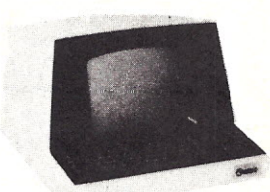
IMMEDIATE DELIVERY

— CALL FOR PRICES —

T.I. 810 as low as $1695



SOROC 120



NEW
LOW
PRICE
$795

INTERTUBE by Intertec

- 12" Display
- 24 x 80 format
- 18-key numeric keypad
- 128 upper/lower case ASCII characters
- Reverse video, blinking
- Complete cursor addressing and control
- Special user-defined control function keys
- Protected and unprotected fields
- Line insert/delete and character insert/ delete editing
- Eleven special line drawing symbols

INTERTUBE II — LIMITED TIME **$784**

BANTAM 550

from PERKIN ELMER

Small in size, light in weight, and low in price — but on top of the list in features and performance.

- Upper and lower case
- Full 24 x 80 format
- Sharp 7 x 10 dot matrix

Get everything you want without paying for things you don't need — List: $996

NOW FROM US AT **$799**

Add $20 for anti-glare screen

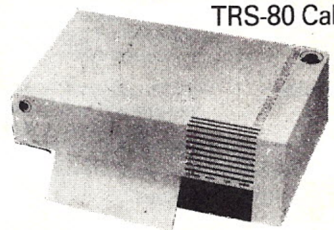
Integral Data Systems, Inc.

IP-125 with 1210 Option*
List $838 OUR PRICE **$754**

IP-225 with Tractor and 1210 Option*, List $984 **$889**

*1210 Option is expanded and compressed print

TRS-80 Cable — $49

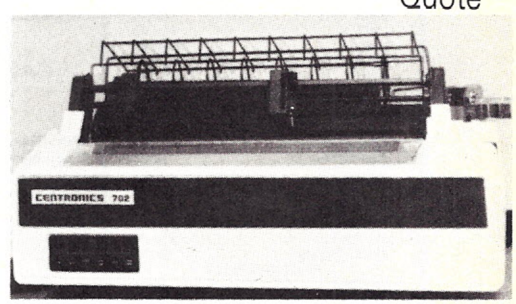


CENTRONICS

779 (60 cps — same as TRS-80 printer OUR PRICE ONLY **$979**

979 with Tractor **$1039**

702 (120 cps, bi-directional)
703 (185 cps, bi-directional) Call for Quote



OTHER VALUES . . .

DECwriter II $1490
RS232 Interface — $45

Teletype 43 Printers as low as $880

— LIMITED TIME OFFER —

HAZELTINE 1500
ONLY **$989**

M6

SHIPPING, HANDLING & INSURANCE: Check with us for specific information.
All prices subject to change without notice and all offers subject to withdrawal without notice

WRITE FOR FREE CATALOG

MiniMicroMart, Inc. 1618 James St., Syracuse, NY 13203 (315) 422-4467 TWX 710 541-0431

R7

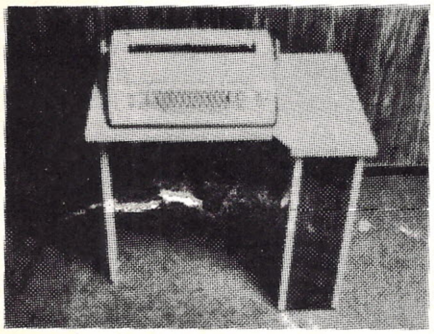
RONDURE COMPANY

2522 BUTLER ST.
DALLAS, TEXAS 75235
214–630-4621

the computer room

TESTED WITH
NEW
ASCII
ELECTRONICS

"works on most micros including TRS-80"



IBM Selectric with ASCII parallel receiver only electronics.
Immediate Delivery—Shipped from inventory.

$1095.00

ASCII SELECTRIC

Printer Mechanism: Heavy duty input/output, Series 745.

Weight: 120 lbs. Dimensions: 29"Hx35"Wx33"D.

Print Speed: (14.8 characters per second)

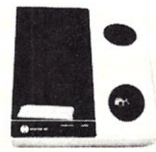
Platen: 15" wide, pin feed or form feed device optional (132 print positions).

Parallel output only—15 characters per second accepts 7 bit ASCII parallel w/strobe & prints on Selectric. The unit still works as a typewriter in off-line mode.

MODEM

NOVATION DC3102A

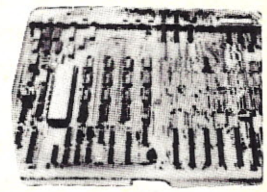
Used
Working



$150.00

RS232 Connection
300 Baud

TI 990/4 Micro



SALE
$225.00

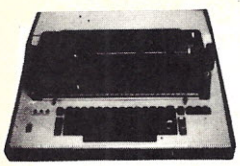
single board cpu
256 WORDS RAM
SOCKETS FOR 1K ROM
DOCUMENTATION

NEW Novation Modems & Couplers

4102A—Acoustic—Originate	$275.00
4103A—Acoustic—Ans/Orig	$295.00
4102—Hard Wire—Orig.	$340.00
4103B—Hard Wire—Orig. & Auto Answer	$385.00
4113B—Hard Wire—Auto Answer Only	$365.00

Write or call for our catalog of many terminals & parts.

DATEL SELECTRIC (IBM Selectric Mechanism)



Specifications:
- Size: 21"Wx21"Dx8"H.
- Power input 115 Volt Hz
- Interface: RS232
- Weight: 54 lbs. (Shipping weight 65 lbs.)
- 15" Carriage
- 15 CPS
- Correspondence code
- Half Duplex
- 132 Print Positions, 10 Pitch

Used, As Is	$295.00
Clean & Working	$495.00

Converted to parallel ASCII printer $900.00
Works on most micros including TRS-80.
Manufacturers Electronic & Mechanical Documentation $40.00

SPECIAL

Printer for your Microcomputer

Will run on serial RS232 port of most micros including TRS-80.

GE TERMINET 300 PRINTER

(used)
(good working condition)

$450.00

Pin feed—9" paper

- Receive only
- ASCII code
- RS-232 interface
- 30 CPS
- Upper & lowercase
- Shipping wt. 75#

WE HAVE FLAT-PACK ACOUSTIC

Modem pickup **$19.50**

NOTE:

Even though market SOLD OUT in catalog--- we've got more.

ASCII Keyboard

(used)
with enclosure

$35.00

A TV Set & a monitor

Sony TV/Monitor

TV Set or TV/Sync Input
11" used

$75.00

ORDERING INFORMATION:
We ship the same day we receive a certified check or money order.
Texas residents add 5% sales tax. Please call if you have a question.
Write for our CATALOG of many parts, terminals, printers, etc.
All items subject to availability. Your money returned if we are out of stock.

SHIPPING INFORMATION:
Modems: $2.00 each; 2 for $4.00 UPS.
Large Items & Parts: Specify Freight or Air Freight Collect
Foreign Orders: Add appropriate freight or postage.
We now take Master Charge and Visa orders, Specify full number, bank number and expiration date.

Reader Service—see page 179

WAMECO

THE COMPLETE PC BOARD HOUSE
EVERYTHING FOR THE S-100 BUSS

- * **FPB-1** FRONT PANEL BOARD
  Hex Displays, IMSAI Replaceable $54.95
- * **FDC-1** FLOPPY DISC CONTROLLER BOARD
  Controls up to 8 Discs $45.00
- * **MEM-1A** 8K BYTE 2102 RAM Board $31.95
- * **MEM-2** 16K BYTE 2114 RAM Board $31.95
- * **CPU-1** 8080A CPU Board
  With Vector Interrupt $31.95
- * **EPM-1** 4K BYTE 1702A EPROM $29.95
- * **EPM-2** 16K or 32K BYTE EPROM
  2708 or 2176 interchangeable $30.00
- * **QMB-9** 9 SLOT MOTHER BOARD
  Terminated $35.00
- * **QMB-12** 12 SLOT MOTHER BOARD
  Terminated $40.00
- * **RTC** REALTIME CLOCK
  Programmable Interrupts $27.95

FUTURE PRODUCTS: 80 CHARACTER VIDEO BOARD,
IO BOARD WITH CASSETTE INTERFACE.

**DEALER INQUIRIES INVITED, UNIVERSITY DISCOUNTS AVAILABLE
AT YOUR LOCAL DEALER**

✓W22 WMC inc. WAMECO INC. 111 GLENN WAY #8, BELMONT, CA 94002 (415) 592-6141

CALIFORNIA COMPUTER SYSTEMS

16K RAM BOARD. Fully buffered addressable in 4K blocks. IEEE standard for bank addressing 2114's
PCBD $26.95
Kit 450 NSEC $259.95

PT-1 PROTO BOARD. Over 2,600 holes 4" regulators. All S-100 buss functions labeled, gold fingers.
PCBD $29.95

PT-2 PROTO BOARD. Similar to PT-1 except set-up to handle solder tail sockets.
PCBD $29.95

SSM

FORMERLY CYBERCOM/SOLID STATE MUSIC.

PB-1 2708 & 2716 Programming Board with provisions for 4K or 8K EPROM. No external supplies require textool sockets. Kit $124.95

CB-1 8080 Processor Board. 2K of PROM 256 BYTE RAM power on/rest Vector Jump Parallel port with status Kit ...... $119.00 PCBD $30.95

MB-6B Basic 8KX8 ram uses 2102 type rams, S-100 buss. Kit 450 NSEC...... $139.95 PCBD...... $26.95

MB-7 16KX8, Static RAM uses μP410 Protection, fully buffered Kit...... $299.95

MB-8A 2708 EROM Board, S-100, 8K8X or 16Kx8 kit without PROMS $75.00 ...... PCBD $28.95

MB-9 4KX8 RAM/PROM Board uses 2112 RAMS or 82S129 PROM kit without RAMS or PROMS $72.00

IO-2 S-100 8 bit parallel /IO port, ⅔ of boards is for kludging. Kit ...... $46.00 PCBD...... $26.95

IO-4 Two serial I/O ports with full handshaking 20/60 ma current loop: Two parallel I/O ports.
Kit ...... $130.00 PCBD...... $26.95

VB-1B 64 x 16 video board, upper lower case Greek, composite and parallel video with software, S-100.
Kit...... $125.00 PCBD...... $26.95

Altair Compatible Mother Board, 11 x 11½ x ⅛".
Board only ...... $39.95. With 15 connectors...... $94.95

Extended Board full size. Board only ...... $ 9.49
With connector $13.45

SP-1 Synthesizer Board S-100
PCBD...... $42.95 KIT...... $135.95

WMC inc. WAMECO INC.

FDC-1 FLOPPY CONTROLLER BOARD will drive shugart, pertek, remic 5" & 8" drives up to 8 drives, on board PROM with power boot up, will operate with CPM (not included).
PCBD $42.95

FPB-1 Front Panel. IMSAI size, hex displays. Byte, or instruction single step.
PCBD $47.50

MEM-1 8KX8 fully buffered, S-100, uses 2102 type rams. PCBD $25.95

QM-12 MOTHER BOARD, 13 slot, terminated, S-100 board only $34.95

CPU-1 8080A Processor board S-100 with 8 level vector interrupt PCBD $26.95

RTC-1 Realtime clock board. Two independent interrupts. Software programmable. PCBD $23.95

EPM-1 1702A 4K Eprom card PCBD $25.95

EPM-2 2708/2716 16K/32K
EPROM CARD PCBD $25.95

QM-9 MOTHER BOARD, Short Version of QM-12.
9 Slots PCBD $30.95

MEM-2 16K x 8 Fully Buffered
2114 Board PCBD $26.95

8080A	$9.95	5101-8P	$ 8.40
8212	2.49	2114 (450 NS) low pwr	7.25
8214	4.49	2114 (250 NS) low pwr	7.99
8224	3.49	2102A-2L	1.50
2708	9.49	2102A-4L	1.20
5101-1P	6.90	4116	8/89.95

MIKOS

(415) 592-1800
P. O. Box 424 • San Carlos, California 94070

✓M39 Please send for IC, Xistor and Computer parts list

**JULY SPECIAL SALE
ON PREPAID ORDERS**
(charge cards not included on this offer)

CCS PROTO BOARDS

PT-1 ...... $27.95 4/$100
PT-2 ...... $27.95 4/$100

**MIKOS PARTS ASSORTMENT
WITH WAMECO AND CYBERCOM PCBDS**

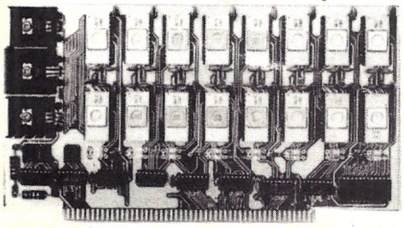
MEM-2 with MIKOS #7 16K ram with L2114 450 NSEC $235.95
MEM-2 with MIKOS #13 16K ram with L2114 250 NSEC $269.95
MEM-1 with MIKOS #1 450 NSEC 8K RAM $119.95
CPU-1 with MIKOS #2 8080A CPU $94.95
MEM-1 with MIKOS #3 250 NSEC 8K RAM $144.95
QM-12 with MIKOS #4 13 slot mother board $89.95
RTC-1 with MIKOS #5 real time clock $54.95
VB-1B with MIKOS #6 video board less molex connectors $99.95
EMP-1 with MIKOS #10 4K 1702 less EPROMS $49.95
EPM-2 with MIKOS #11 16-32K EPROMS less EPROMS $59.95
QM-9 with MIKOS #12 9 slot mother board $79.95
FPB-1 with MIKOS #14 all parts for front panel $134.95

MIKOS PARTS ASSORTMENTS ARE ALL FACTORY PRIME PARTS. KITS INCLUDE ALL PARTS LISTED AS REQUIRED FOR THE COMPLETE KIT LESS PARTS LISTED. ALL SOCKETS INCLUDED.

VISA or MASTERCHARGE. Send account number, interbank number, expiration date and sign your order. Approx. postage will be added. Check or money order will be sent post paid in U.S. If you are not a regular customer, please use charge, cashier's check or postal money order. Otherwise there will be a two-week delay for checks to clear. Calif. residents add 6% tax. Money back 30 day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. **$10 minimum order. $1.50 service charge on orders less than $10.00.**

Reader Service—see page 179

16K EPROM CARD-S 100 BUSS



$59.95 KIT

OUR BEST SELLING KIT!

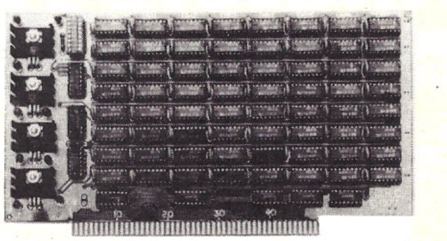
USES 2708's!

Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. All parts (except 2708's) are included. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

OUR 450NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT

ASSEMBLED AND FULLY TESTED ADD $25

8K LOW POWER RAM KIT-S 100 BUSS
250 NS SALE!



ADD $5 FOR 250NS!

$129 KIT

(450 NS RAMS!)

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

KIT FEATURES:

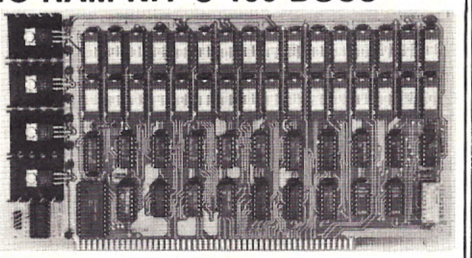
1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

Blank PC Board w/Documentation **$29.95**
Low Profile Socket Set...**13.50**
Support IC's (TTL & Regulators) **$9.75**
Bypass CAP's (Disc & Tantalums) **$4.50**

ASSEMBLED AND FULLY BURNED IN ADD $30

16K STATIC RAM KIT-S 100 BUSS

$295 KIT



FULLY STATIC, AT DYNAMIC PRICES

WHY THE 2114 RAM CHIP?

We feel the 2114 will be the next industry standard RAM chip (like the 2102 was). This means price, availability, and quality will all be good! Next, the 2114 is FULLY STATIC! We feel this is the **ONLY** way to go on the S-100 Buss! We've all heard the HORROR stories about some Dynamic Ram Boards having trouble with DMA and FLOPPY DISC DRIVES. Who needs these kinds of problems? And finally, even among other 4K Static RAM's the 2114 stands out! Not all 4K static Rams are created equal! Some of the other 4K's have **clocked** chip enable lines and various timing windows **just as critical** as Dynamic RAM's. Some of our competitor's 16K boards use these "tricky" devices. But not us! The 2114 is the ONLY logical choice for a trouble-free, straightforward design.

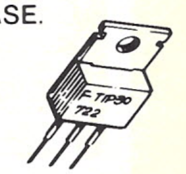
KIT FEATURES:

1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 2 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA—**$33**
LOW PROFILE SOCKET SET—**$12**
SUPPORT IC'S & CAPS—**$19.95**
ASSEMBLED & TESTED—**ADD $30**
2114 RAM'S—**8 FOR $69.95**

COMPLEMENTARY POWER TRANSISTORS

SILICON NPN AND PNP. TO-220 CASE.
VCEO - 40V PD - 30 WATTS
FOR AUDIO POWER AMPS, ETC.



TIP29 - NPN
TIP30 - PNP

YOUR CHOICE
3 FOR $1

16K DYNAMIC RAM CHIP

16K X 1 Bits. 16 Pin Package. Same as Mostek 4116-4. 250 NS access. 410 NS cycle time. Our best price yet for this state of the art RAM. 32K and 64K RAM boards using this chip are readily available. These are new, fully guaranteed devices by a major mfg. **VERY LIMITED STOCK!**

8 FOR $89.95

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

450 NS!
2708 EPROMS

Now full speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4-1702 A's in one package.

~~$15.75 ea.~~ **$9.95** ~~4 FOR $50.00~~

PRICE CUT

NATIONAL SEMICONDUCTOR ***NEW!***
CAR CLOCK MODULE - #MA6008

$6.99 each

Originally used by HYGAIN to indicate time and channel on an expensive C.B. Mini size, self contained module. Not a Kit. Four digits plus flashing indicator for seconds. Includes MM5369 and 3.58 MHZ crystal for super accurate time base. With hookup data.

INCLUDES CRYSTAL TIMEBASE! WORKS ON 12 VDC!

MFGR's CLOSEOUT LIMITED QTY.

EXPERIMENTER'S HEATING PLATE

Large Manufacturers Surplus. 5¼x10½ in. Made of 3/8 in. tempered glass with heating element laminated on back. Works off 120 VAC. Protected by thermostat and two thermal fuses. Rated 120 Watts. Use for any heating applications. Perfect for heating ferric chloride to increase PC Board etching efficiency. Units are brand new, non-submersible.

WHILE THEY LAST—$2.99 each

MALLORY COMPUTER GRADE CAPACITOR
30,000 MFD 15 WVDC
Small: 3 x 2 Inches
$1.99 ea. 3 For $4.99

New! REAL TIME Computer Clock Chip
N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

Z-80 PROGRAMMING MANUAL

By MOSTEK, or ZILOG. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A MUST reference manual for any user of the Z-80. 300 pages. Just off the press. **$12.95**

GENERAL INSTRUMENT
FULL WAVE BRIDGE
4 AMP 600 PIV
LOOK!
3/4 IN. SQUARE - WITH LUGS
75¢ ea. 3 FOR $2 #LM-1

"THE COLOSSUS"
FAIRCHILD SUPER JUMBO LED READOUT

A full .80 inch character. The biggest readout we have ever sold! Super efficient. Compare at up to $2.95 each from others!
YOUR CHOICE
FND 843 Common Anode
FND 850 Common Cathode **$1.49 ea (6 for $6.95)**

Digital Research Corporation
D20
(OF TEXAS)
P.O. BOX 401247K GARLAND, TEXAS 75040 • (214) 271-2461

TERMS: Add 30¢ postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa, MasterCharge, and American Express cards. Tex. Res. add 5% Tax. Foreign orders (except Canada add 20% P & H. 90 Day Money Back Guarantee on all items.

TEST EQUIPMENT
CONNECTORS - WIRE
ELECTRONIC COMPONENTS

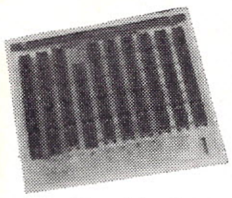
WALLEN ELECTRONICS CO. INC. Tel: (617) 588-6440-6441
108 SAWTELL AVE., BROCKTON, MA. 02402

Power Supplies REGULATED COMPUTER

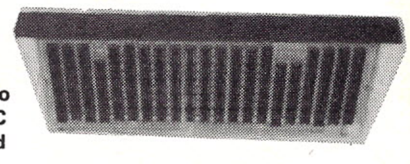
5 volts @ 8 amps + 12 volts. 2 amps + 6 volts @ 75 MA. Ideal for Micro and Mini computers. These units have been removed from equipment and checked out and guaranteed. Power supply has a 3-wire line cord and fused. Specs: 10½"x5½"x4½". Shipping weight 16 lbs.

37.50

2 FOR 70.00



Wire Wrap Boards

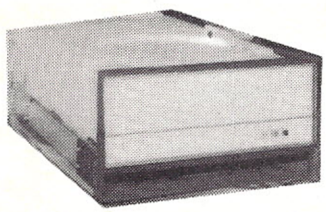


These boards are pre-wired and removed from equipment. Easy to unwrap for setting up your own board. contains mostly 14-pin IC sockets with individual pin connections. Each board has VCC and ground planes.

Smaller board measures 6½"x6" and has 40 to 50 sockets. PRICE $11.00 ea. 2/$20.00
Larger board measures 13½"x6" and has 75 to 100 sockets. PRICE $17.50 ea. 2/$33.00

Diablo System DISK DRIVE

SERIES 40 MODEL 43

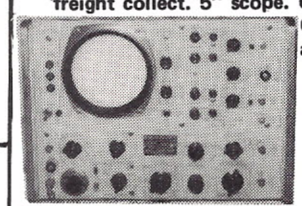


100 tracks per inch, total capacity of 50 megabits, w/Model 429 power supply, sector counter, 24 sectors, 1 fixed disc, 1 removable disc, average access time 38 ms, PPM: 2400, dimensions: 10 5/16" high, fits in standard rack, equipped with full extension slides. Excellent used condition.

$2495

HEWLETT-PACKARD Model 175A OSCILLOSCOPES

These scopes have a 50 Mhz band-width and have 2 plug-ins, a 1781B Delay generator and a model 1755A Dual trace vertical amplifier. Dimensions: 13"x17"x25", weight 71 lbs. shipped freight collect. 5" scope. Used.



Checked out and operating.

$339

TRANSFORMERS ISOLATION STEP-DOWN TYPE

Primary: 230/115V. 50/60 CPS.
Secondary: VA output 250V.

$13.95 EACH

ROTRON WHISPER FANS

Unused, Model Rotron MU3A1, 230V, AC, 14 watts, 50/60 hz, guaranteed. 4½"x4½"x1½".

$8.95

SEMI-CONDUCTOR COOLERS
(HEAT SINKS)

ALUMINUM BLANKS

NOT DRILLED EACH **1.75**

5"x4¾"x1¼"

Anodized Aluminum For TO-3 or T0-66 TRANSISTORS

CHASSIS MOUNTING **20¢** EACH

1 7/8"x1 1/8"

For TO-5 Transistors (Gear Type)

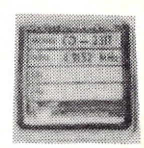
$15

PACKAGE

SOLD IN PKG. OF 100s

CRYSTAL OSCILLATORS

Vectron type CO-231T crystal freq. 4.9152, MHZ w/tuning option for accuracy of .0001%. 1½"x1½"x½". R.F.E.



13.95

FM SIGNAL GENERATOR

MEASUREMENTS MODEL 560 FM

Frequency 25 mhz to 80 mhz, 130-175 mhz. Dimensions: 10"x10"x16", weight: 28 lbs. Shipped freight collect. Used. Checked out and operating.

$289

INCANDESCENT READOUT ASSEMBLY

Readouts assembled of the 710 series modules. Character 1" high w/lamps. Type No. 344. By Dialco.

2.75 EACH

Minimum order $25.00. Items offered subject to prior sale. FOB, Brockton, Mass. Money order or check w/order. Shipping and handling add 5%. Shipments by parcel post or UPS. No CODs. Mass. residents add 5% sales tax.

10-DAY FREE TRIAL

Send for our FREE Catalog

COMPUMART'S PET WAREHOUSE

We have more PETs than anyone, call for quick delivery.

$100 worth of FREE accessories with purchase of 16K & 32K PETS!

AS LOW AS $595

4K - Keyboard C	$ 595
8K - Keyboard C	$ 795
16K - Keyboard B	$ 995
16K - Keyboard N	$ 995
32K - Keyboard C	$1195
32K - Keyboard B	$1295
32K - Keyboard N	$1295

C — calculator keyboard (only version with tape deck)
B — large business keyboard without graphics symbols
N — large keyboard with graphics symbols

PET ACCESSORIES

Commodore Dual Floppy Disk Drive	$1295.00
New! PET Terminal Package	$69.00
Second Cassette — from Commodore	$95.00
Commodore PET Service Kit	$30.00
Beeper - Tells when tape is loaded	$24.95
Petunia - Play music from PET	$29.95
Video Buffer - Attach another CRT	$29.95
Combo - Petunia and Video Buffer	$49.95
New Serial Printer Interface for PET	$79.95
Integral Data Printer w/new interface	$878.95
PET - Compatible Selectric in Desk	$895.00
TTY KSR-33 Screen Printer for PET SALE!	$395.00
Originate/Answerback Modem for PET	$320.00
Bi-directional RS-232 Interface	$280.00
Betsi 4-slot S-100 Motherboard SALE!	$119.00
S-100 PET Interface was $289.00 SALE	$99.00

FREE SANYO MONITOR

A $169 VALUE

when you buy the

SORCERER

$895

- Z-80 Processor
- Full-size ASCII keyboard
- User definable characters
- Up to 32K on-board RAM
- Interchangeable ROM PACs
- 30 x 64 display
- Resident 4K Monitor ROM
- Dual Cassette I/O
- Serial and Parallel I/O
- Complete Operators Manual
- Only $895

Top value in our catalog, simply add a monitor and tape decks to complete the system. Can be used as a terminal. Later ROM PACs will offer Z-80 Assembly Language, APL, PILOT, Word Processing and a DOS PAC for disk-based FORTRAN and COBOL. What a system for $895.

Sorcerer 8K	$895
Sorcerer 16K	$1150
Sorcerer 32K	$1395
16K Expansion	$299

apple II

$200 FREE ACCESSORIES

Buy a 48K Apple II, mention this ad and take $200 in accessories free (if ordered together). This offer is good for $150 on 32K and $100 on 16K Apple II's. Now you can enjoy more of the best for less.

16K Apple II — $1195 (take $100 in free accessories)
32K Apple II — $1345 (take $150 in free accessories)
48K Apple II — $1495 (take $200 in free accessories)

Apple II Accessories

Centronics Printer Interface	$225
Disk and Controller	$595
Second Disk Drive	$495
Parallel Printer Card	$180
Communications Card	$225
Hi-Speed Serial Card	$195
Firmware Card	$200
Hobby/Proto Card	$24
Microverter RF Mod.	$35
Sanyo M2544 Recorder	$55

SPECIAL OF THE MONTH

FREE MINI/MICRO CATALOG

We've just published our brand new 48 page catalog and its full of great mini and micro computer buys.

The microcomputer section is packed with illustrations and text presenting all that's new in personal computing. All the items you see in this ad are described in much greater detail. Featured is the Apple II and the PET along with the newest PET accessories and disk.

Other popular computers, computer peripherals, supplies and books are described, along with some special deals you can get only if you have this catalog.

To get your free catalog just write us at the address listed at the bottom of this ad. Please be sure to include the department number to insure speedy delivery.

50% OFF CENTRONICS PRINTERS

While they last.

Refurbished, excellent working condition. 90 day warranty. 10 day return privilege of course.

Model	Specifications	Centronics List	NCE Price
CENTRONICS 700	60 cps. 132 64 5 x 7 chars tractor feed	$1.520	$ 825
CENTRONICS 701	Same as 700. bidirectional	1.815	1.045
CENTRONICS 703	Bidirectional. 180 cps. 132 col. 64 7 x 7 chars tractor feed	2.805	1.695
CENTRONICS 779	60 100 cps. 80 132 col. 64 5 x 7 chars	1.250	625
CENTRONICS 780	60 cps. 80 col. 64 5 x 7 chars	1.905	1.095
CENTRONICS 781	Bidirectional. 60 cps. 80 col. 64 5 x 7 chars tractor feed	1.980	1.125
CENTRONICS 761-1 (KSR)	Keyboard send receive. bidirectional 60 cps. 132 column. 64 7 x 7 chars tractor feed. 110 300 1200 baud	1.850	1.025
CENTRONICS 761-5 (RO)	Same as 761-1. receive only	1.750	945
		NCE Reg. Price	
CENTRONICS 101	165 cps. 132 col. 64 5 x 7 chars tractor feed	$1.500	1.125

Operator's Manual Included. For Technical Manual, add $15.00

Hazeltine 1400

IN STOCK NOW Only $699.00

Now a 5th Generation in Terminals: Immediate Delivery

Hazeltine 1410 with numeric keypad	$835
Hazeltine 1500 full of features SALE!	$1099
Hazeltine 1510 with buffer logic	$1149
Hazeltine 1520 with printer interface	$1499

call or write for more information

SALE!

LOW-COST PRINTER FOR PET

This is a completely refurbished Teletype KSR-33 terminal ready to attach to your PET's interface to use as a printer

INTRODUCTORY PRICE $395

SALE!

TI PROGRAMMER

Hexadecimal, octal, decimal calculator converter

Reg. $59.95 NOW **$49.95**

Datel/Selectric Table-top Terminals

- Local, remote or wait status selection
- Computer controllable terminal functions
- Remote interrogation of terminal status
- Interrupt and reverse break
- Interface RS232C BCD code

SALE! ~~$395~~ $295 AS IS

CAT COUPLER

New 300 baud Originate/Answerback Acoustic Coupler. Looks good, works great and sale priced at **$189.**

NOW IN STOCK

SYM BASIC $159

KIM-1

The original fully expandable single board computer

SALE Assembled **$169.95**

with power supply **$214.95**

EXPAND YOUR KIM SYSTEM:

KIMSI

KIM to S-100 Board

Assembled	$169
Connector Set	15

MICRO TECH. POWER SUPPLY
Assembled $34.95

KIM ENCLOSURE

Give your KIM a professional appearance. No alterations required, sturdy Kydex 100 Black

$23.50

NCE/CompuMart, Inc.

1250 North Main Street, Department KB79
P.O. Box 8610 Ann Arbor, Michigan 48107

10 DAY RETURN PRIVILEGE

(313) 994-3200

Just say charge it! Which card?

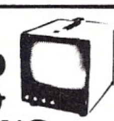
SUPER SALE

List $279.95

$99.00

S-100 MPA

S-100 MPA gives your PET complete control of the S-100 bus (even DMA). Get an assembled unit at kit price.



SALE **$169** ~~$240~~

SANYO MONITOR

VM4209 9" quality monitor

• Michigan residents add 4% sales tax • Add 4% shipping and handling for all domestic orders • Foreign orders (except Canada) add 10% additional handling (U.S. currency only) • P.O.'s accepted from D & B rated companies — shipment contingent upon receipt of signed purchase order • Phones open from 9:00 a.m. to 7:00 p.m. EST Monday-Friday, 10:00 a.m. to 5:00 p.m. Saturdays

• Open accounts invited — call for credit application • Most items in stock for immediate shipment — call for delivery quotation • Sorry, no C.O.D.'s • All prices subject to change without notice • In the Ann Arbor area? Retail store open 11:00 a.m. to 7:00 p.m. Tuesday-Friday, 10:00 a.m. to 5:00 p.m. Saturdays (Closed Sunday and Monday)

HICKOK LX303 $74.95

HICKOK LX303 $74.95

HICKOK LX303 $74.95

HICKOK LX303 $74.95

HICKOK LX303 $74.95

HICKOK LX303 $74.95

HICKOK LX303 $74.95

HICKOK LX303

\$74.95*

.5%, 3½ digit 19 Range DVM. ½" LCD displays runs 200 hrs on 1 battery. 10 Meg Ohm Input. 1 yr. guarantee, made in U.S.A., test leads included.

Available Accessories

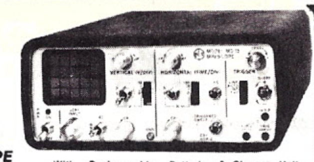
- **RC-3** 115V AC Adapter **$7.50**
- **CC-3** Deluxe Padded Vinyl Carrying Case **$7.50**
- **VP-10** X10 DCV Probe Adapter/ Protector 10Kv **$14.95**
- **VP-40** 40Kv DC Probe **$35.00**
- **CS-1** 10 Amp Current Shunt **$14.95**

*FREE

Just for Asking.

FREE BATTERY with your meter.

NLS



$318.

MS-15

MINISCOPE

With Rechargeable Batteries & Charger Unit

- 15 megahertz bandwidth.
- External and internal trigger.
- Time base — .1 microsec. to 0.5 Sec/div - 21 settings . ± 3%.
- Battery or line operation.
- Automatic & line sync modes.
- Power consumption < 15 watts.
- Vortical Gain — .01 to 50 V/div - 12 settings ± 3%.
- Viewing area 1.1" x 1.35".
- Case size 2.7"H x 6.4"W x 7.5"D, 3 pounds.
- Parts & Labor guaranteed 1 year
- 10 to 1, 10 meg probe .... $ 27.
- Leather carrying case .... $ 45.

PROBE 1¢

PROBE 1¢ with the purchase of SCOPE and the MENTION of this MAGAZINE

● **MS-215 Dual Trace Version of MS-15 $435.**

UNGARmatic®

Controlled Soldering Station

THREAD-TOGETHER MODULAR DESIGN FOR QUICK, ON-LINE HEATER OR TIP CHANGE

- Available in 3 preset temperatures; 600°F., 700°F. or 800°F. for any application.
- Closed loop, non-magnetic control.
- Low voltage system; 3 wire grounded.
- Biomechanical designed handle with cool grip for operator comfort. Cord is super flexible 3 wire grounded, heat resistant.
- Large capacity snap-on tray and sponge; removable for optional placement.
- Long life interchangeable tips, iron clad, chrome plated, pre-tinned.
- Designed for use on sensitive components.

COMPLETE STATION AND REPLACEABLE HEATER SELECTION GUIDE

Temperature	Complete Station	Controlled Heater No.
600°F.	50T6	#76
700°F.	50T7	#77
800°F.	50T8	#78

Each of the above stations include: #70B Power Supply with on/off switch, indicator light, and 3 wire power cord; #71 Handle with 3 wire heat resistant secondary cord; #89 Tray and Sponge; #72 Iron Holder; Controlled Heater with #87 Screwdriver tip 1/16".

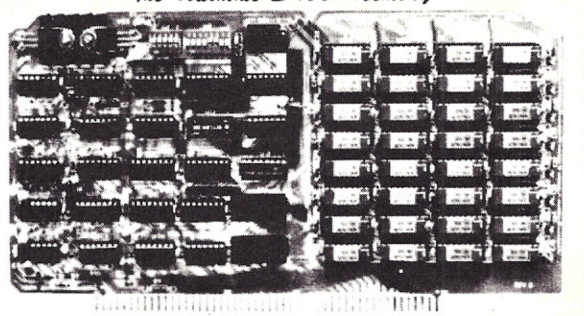
#95 TIP ADAPTER Tip Adapter #95 for special micro applications. This Adapter is designed for the use of 1/8" thread-in Princess Nibs.

Controlled Heaters: $14.95 Each. Tips: $2.25 Each. Adapter: $1.30 Each

List $59.50 SALE PRICE $49.95

SALE SD EXPANDORAM

The Ultimate S-100 Memory



The EXPANDORAM is available in versions from 16K up to 64K, so for a minimum investment you can have a memory system that will grow with your needs. This is a dynamic memory with the invisable on-board refresh, and IT WORKS!

- Bank Selectable
- Phantom
- Power 8VDC, ± 16VDC, 5 Watts
- Lowest Cost Per Bit
- Uses Popular 4116 RAMS
- PC Board is doubled solder masked and has silk-screen parts layout.
- Extensive documentation clearly written
- Complete Kit includes all Sockets for 64K
- Memory access time: 375ns, Cycle time: 500ns.
- No wait states required.
- 16K boundries and Protection via Dip Switches
- Designed to work with Z-80, 8080, 8085 CPU's.

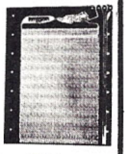
EXPANDO 64 KIT (4116)

- **16K** **$245.00**
- **32K** **$310.00**
- **48K** **$375.00**
- **64K** **$440.00**

8803 MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS

Vector

- Kit includes 12 tantalum capacitors for +5, +12, —12 buses and insulated mounting spacers.
- Wiring side shown. Component side bare epoxy glass with white markings for component locations.
- G10 epoxy glass board with 2 ounce copper, solder plated and .038 diameter holes for leads.
- Solder mask with solder windows on etched circuits to avoid accidental short circuits.
- Mounts 11 receptacles with 100 contacts (2 rows) on .125 centers with .250 row spacing. Vector part number R681-2; or mounts 10 receptacles plus interconnections to smaller mother board for expansion.
- Includes etched circuits and instructions for option of active, pull-up, or floating terminations.
- Large buses: +5V and GND (10 AMPS), ± 12V or 16V (7 AMPS). Current ratings are per MIL-STD-275 with 10°C rise.
- Fits in Vector-pak enclosures.
- Fits in IMSAI 8080 microcomputer as expander board.



Price: $29.50

Vector Plugboards

8800V

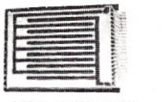
Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware. 5.3" x 10" x 1/16"

1-4	5-9	10-24
$19.95	$17.95	$15.96

8801-1

Same as 8800V except plain; less power buses & heat sink.

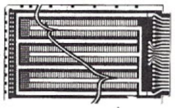
1-4	5-9	10-24
$14.95	$13.46	$11.96



3682 9.6" x 4.5" **$10.97**

3682-2 6.5" x 4.5" **$9.81**

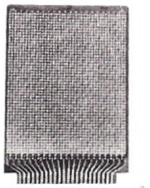
Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156



3677 9.6" x 4.5" **$10.90**

3677-2 6.5" x 4.5" **$9.74**

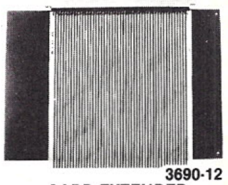
Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156



3662 6.5" x 4.5" **$7.65**

3662-2 9.6" x 4.5" **$11.45**

P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156



3690-12

CARD EXTENDER

Card Extender has 100 contacts 50 per side on .125 centers-Attached connector-is compatible with S-100 Bus Systems. **$25.83**

3690 6.5" 22/44 pin .156 ctrs. Extenders $13.17

1/16 Vector BOARD

.042 dia holes on 0.1 spacing for IC's

		PRICE	
Phenolic			
PART NO.	SIZE	1-9	10-19
64P44XXXP	4.5x6.5"	$1.56	$1.40
169P44XXXP	4.5x17"	$3.69	$3.32
Epoxy Glass			
64P44	4.5x6.5"	$1.79	$1.61
84P44	4.5x8.5"	$2.21	$1.99
169P44	4.5x17"	$4.52	$4.07
169P84	8.5x17"	$8.03	$7.23

TRS-80/APPLE

MEMORY EXPANSION KITS

4116's RAMS

(16Kx1 200ns)

8 for $69.00

2708

8K 450 ns

EPROM

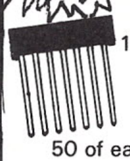
FACTORY PRIME

$ 9.00 ea.

25 + *Call For Price*

WOW

14 & 16 PIN GOLD 3 LEVEL WIRE WRAP SOCKETS



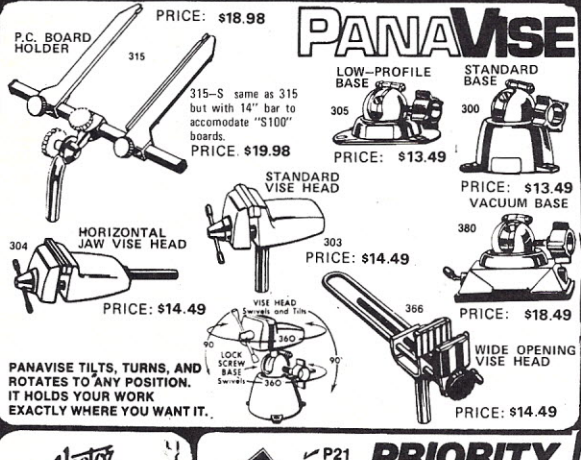
14 - G3 100 for **$30.00**

16-G3 100 for **$30.00**

50 of each for **$32.00**

Sockets are End & Side stackable, closed entry

PanaVise



PANAVISE TILTS, TURNS, AND ROTATES TO ANY POSITION. IT HOLDS YOUR WORK EXACTLY WHERE YOU WANT IT.

2114N-3L

4096 - BIT (1024x4) 300 ns Low Power Static Ram

8 for \$50.00

ORDER TOLL FREE

1-800-423-5633

except CA., AK., HI., Call (213) 894-8171

2102LPC

450ns Low Power RAMS

$1.00 Ea. in lots of 25

2102LHPC

250ns Low Power RAMS

$1.25 Ea. in lots of 25

TMS 4044/ MM 5257-3L

4096x1 300 ns Low Power Static Ram

8 for \$50.00

Vector

WRAP POST

for .042 dia. holes (all boards on this page)

- **T44/C** pkg. 100 .. **$ 2.34**
- **T44/M** pkg. 1000 ..... **$14.35**
- **A-13** hand installing tool .... **$ 2.94**

P21 **PRIORITY ONE ELECTRONICS** ©

16723K Roscoe Blvd. Sepulveda, CA 91343

VISA · master charge

Terms: Visa, MC, BAC, Check, Money Order, C.O.D. U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Prepaid U.S. orders less than $75.00 include 5% shipping and handling, minimum $2.50. Excess refunded. Just in case . . . please include your phone no. Prices subject to change without notice.

We will do our best to maintain prices thru June 1979.

phone orders welcome (213) 894-8171, (800) 423-5633

OEM and Institutional inquiries invited.

2716

5 volt only 16K EPROM

\$50.00

HICKOK LX303 $74.95

HICKOK LX303 $74.95

DIODES/ZENERS

QTY.				
	1N914	100v	10mA	.05
	1N4005	600v	1A	.08
	1N4007	1000v	1A	.15
	1N4148	75v	10mA	.05
	1N4733	5.1v	1 W Zenner	.25
	1N4749	24v	1W	.25
	1N753A	6.2v	500 mW Zener	.25
	1N758A	10v	"	.25
	1N759A	12v	"	.25
	1N5243	13v	"	.25
	1N5244B	14v	"	.25
	1N5245B	15v	"	.25
	1N5349	12v	3W	.25

SOCKETS/BRIDGES

QTY.					
	8-pin	pcb	.16	ww	.35
	14-pin	pcb	.20	ww	.40
	16-pin	pcb	.25	ww	.45
	18-pin	pcb	.30	ww	.95
	20-pin	pcb	.35	ww	1.05
	22-pin	pcb	.40	ww	1.15
	24-pin	pcb	.45	ww	1.25
	28-pin	pcb	.50	ww	1.35
	40-pin	pcb	.55	ww	1.45
	Molex pins	.01	To-3 Sockets		.35
	2 Amp Bridge		100-prv		.95
	25 Amp Bridge		200-prv		1.50

TRANSISTORS, LEDS, etc.

QTY.			
	2N2222M	(2N2222 Plastic .10)	.15
	2N2222A		.19
	2N2907A	PNP	.19
	2N3906	PNP (Plastic)	.19
	2N3904	NPN (Plastic)	.19
	2N3054	NPN	.55
	2N3055	NPN 15A 60v	.60
	T1P125	PNP Darlington	1.95
	LED Green,	Red, Clear, Yellow	.19
	D.L.747	7 seg 5/8" High com-anode	1.95
	MAN72	7 seg com-anode (Red)	1.25
	MAN3610	7 seg com-anode (Orange)	1.25
	MAN82A	7 seg com-anode (Yellow)	1.25
	MAN74	7 seg com-cathode (Red)	1.50
	FND359	7 seg com-cathode (Red)	1.25

9000 SERIES

QTY.			QTY.		
	9301	.85		9322	.65
	9309	.50		9601	.30
				9602	.45

MICRO's, RAMS, CPU's, E-PROMS

QTY.		
	8T13	2.50
	8T23	2.50
	8T24	3.00
	8T97	1.75
	74S188	3.00
	1488	1.25
	1489	1.25
	1702A	4.50
	AM 9050	4.00
	ICM 7207	6.95
	ICM 7208	13.95
	MPS 6520	10.00
	MM 5314	4.00
	MM 5316	4.50
	MM 5387	3.50
	MM 5369	2.95
	TR 1602B	3.95
	UPD 414	4.95
	Z 80 A	22.50
	Z 80	17.50
	Z 80 P10	10.50
	2102	1.45
	2102L	1.75
	2107B-4	4.95
	2114	9.50
	2513	6.25
	2708	11.50
	2716 D.S.	34.00
	2716 (5v)	69.00
	2758 (5v)	26.95
	3242	10.50
	4116	11.50
	6800	13.95
	6850	7.95
	8080	7.50
	8085	22.50
	8212	2.75
	8214	4.95
	8216	3.50
	8224	4.25
	8228	6.00
	8251	7.50
	8253	18.50
	8255	8.50
	TMS 4044	9.95

– T T L –

4-1-79

QTY.			QTY.			QTY.			QTY.		
	7400	.20		7492	.45		74H20	.25		74LS76	.70
	7401	.20		7493	.35		74H21	.25		74LS86	.95
	7402	.20		7494	.75		74H22	.40		74LS90	.85
	7403	.20		7495	.60		74H30	.30		74LS93	.85
	7404	.20		7496	.80		74H40	.35		74LS96	2.00
	7405	.35		74100	1.15		74H50	.30		74LS107	.90
	7406	.25		74107	.35		74H51	.30		74LS109	1.50
	7407	.55		74121	.35		74H52	.20		74LS123	1.95
	7408	.20		74122	.55		74H53	.25		74LS138	2.00
	7409	.25		74123	.55		74H55	.25		74LS151	.95
	7410	.20		74125	.45		74H72	.35		74LS153	1.15
	7411	.25		74126	.45		74H74	.35		74LS157	1.15
	7412	.25		74132	.75		74H101	.95		74LS160	1.15
	7413	.45		74141	.90		74H103	.55		74LS164	2.90
	7414	.75		74150	.85		74H106	1.15		74LS193	2.00
	7416	.25		74151	.95		74L00	.30		74LS195	1.15
	7417	.40		74153	.95		74L02	.30		74LS244	2.90
	7420	.25		74154	1.15		74L03	.35		74LS259	1.50
	7426	.25		74156	.70		74L04	.40		74LS298	1.50
	7427	.25		74157	.65		74L10	.30		74LS367	1.95
	7430	.20		74161/9316	.75		74L20	.45		74LS368	1.25
	7432	.30		74163	.85		74L30	.55		74LS373	2.50
	7437	.20		74164	.75		74L47	1.95		74S00	.45
	7438	.30		74165	1.10		74L51	.65		74S02	.45
	7440	.20		74166	1.75		74L55	.85		74S03	.35
	7441	1.15		74175	.90		74L72	.65		74S04	.35
	7442	.55		74176	.95		74L73	.70		74S05	.45
	7443	.45		74177	1.10		74L74	.75		74S08	.45
	7444	.45		74180	.95		74L75	1.05		74S10	.45
	7445	.75		74181	2.25		74L85	2.00		74S11	.45
	7446	.70		74182	.75		74L93	.75		74S20	.35
	7447	.70		74190	1.25		74L123	1.95		74S22	.55
	7448	.50		74191	1.25		74LS00	.40		74S40	.30
	7450	.25		74192	.75		74LS01	.40		74S50	.30
	7451	.25		74193	.85		74LS02	.45		74S51	.35
	7453	.20		74194	.95		74LS03	.45		74S64	.15
	7454	.25		74195	.95		74LS04	.45		74S74	.70
	7460	.40		74196	.95		74LS05	.45		74S112	.60
	7470	.45		74197	.95		74LS08	.45		74S114	.85
	7472	.40		74198	1.45		74LS09	.45		74S133	.85
	7473	.25		74221	1.50		74LS10	.45		74S140	.75
	7474	.30		74298	1.50		74LS11	.45		74S151	.95
	7475	.35		74367	1.35		74LS20	.45		74S153	.95
	7476	.40		75491	.65		74LS21	.45		74S157	.98
	7480	.75		75492	.65		74LS22	.45		74S158	.80
	7481	.85		74H00	.20		74LS32	.50		74S194	1.50
	7482	.95		74H01	.30		74LS37	.45		74S196	2.00
	7483	.95		74H04	.30		74LS38	.65		74S257 (8123)	2.50
	7485	.75		74H05	.25		74LS40	.70		8131	2.75
	7486	.55		74H08	.35		74LS42	.95			
	7489	1.05		74H10	.35		74LS51	.75			
	7490	.55		74H11	.25		74LS74	.95			
	7491	.70		74H15	.45		74LS75	1.20			

C MOS

QTY.			QTY.			QTY.			QTY.		
	4000	.15		4017	.75		4034	2.45		4069/74C04	.45
	4001	.20		4018	.75		4035	.75		4071	.25
	4002	.25		4019	.35		4037	1.80		4081	.30
	4004	3.95		4020	.85		4040	.75		4082	.30
	4006	.95		4021	.75		4041	.69		4507	.95
	4007	.25		4022	.75		4042	.65		4511	.95
	4008	.75		4023	.25		4043	.50		4512	1.50
	4009	.35		4024	.75		4044	.65		4515	2.95
	4010	.35		4025	.25		4046	1.25		4519	.85
	4011	.30		4026	1.95		4047	2.50		4522	1.10
	4012	.25		4027	.35		4048	1.25		4526	.95
	4013	.40		4028	.75		4049	.65		4528	1.10
	4014	.75		4029	1.15		4050	.45		4529	.95
	4015	.75		4030	.30		4052	.75		MC14409	14.50
	4016	.35		4033	1.50		4053	.95		MC14419	4.85
							4066	.75		74C151	2.50

I^2L, LINEARS, REGULATORS, ETC.

QTY.			QTY.			QTY.		
	MCT2	.95		LM320K24	1.65		LM373	3.95
	8038	3.95		LM320T5	1.65		LM377	3.95
	LM201	.75		LM320T12	1.65		78L05	.75
	LM301	.45		LM320T15	1.65		78L12	.75
	LM308	.65		LM323K	5.95		78L15	.75
	LM309H	.85		LM324	1.25		78M05	.75
	LM309 (340K-5)	1.50		LM339	.75		LM380 (8-14 Pin)	1.19
	LM310	.85		7805 (340T5)	1.15		LM709 (8-14 Pin)	.45
	LM311 (8-14 Pin)	.75		LM340T12	.95		LM711	.45
	LM318	1.50		LM340T15	.95		LM723	.40
	LM320H6	.79		LM340T18	.95		LM725	2.50
	LM320H15	.79		LM340T24	.95		LM739	1.50
	LM320H24	.79		LM340K12	1.25		LM741 (8-14)	.45
	7905 (LM320K5)	1.65		LM340K15	1.25		LM747	1.10
	LM320K12	1.65		LM340K18	1.25		LM1307	1.75
	LM320K15	1.65		LM340K24	1.25		LM1458	.65
							LM3900	.95
							LM75451	.65
							NE555	.45
							NE556	.85
							NE565	1.15
							NE566	1.25
							NE567	.95
							TA7205	6.95
							76477	2.95
							95H90	9.95

INTEGRATED CIRCUITS UNLIMITED

7889 Clairemont Mesa Blvd., San Diego, California 92111
Out of State 1-800-854-2211 Cable Address:ICUSD Telex: 697-827
(714) 278-4394 California Residents 1-800-542-6239

NAME ____________________

STREET ADDRESS ____________________

CITY ____________ STATE ____________ ZIP ____________

PHONE ____________ CHARGE CARD # AE Visa BA MC ____________ EXP. DATE ______

C.O.D. _____ WILL CALL _____ UPS _____ POST _____ NET 10th OF THE MONTH _____ PO # ____________

ALL ORDERS SHIPPED PREPAID – NO MINIMUM – COD ORDERS ACCEPTED – ALL ORDERS SHIPPED SAME DAY
OPEN ACCOUNTS INVITED – California Residents add 6% Sales Tax. PRICES SUBJECT TO CHANGE WITHOUT NOTICE.
24 Hour Phone Service – We accept American Express / Visa / BankAmericard / Master Charge

SPECIAL DISCOUNTS

Total Order	Deduct
$35-$99	10%
$100-$300	15%
$301-$1000	20%

We have what you want... when you want it.

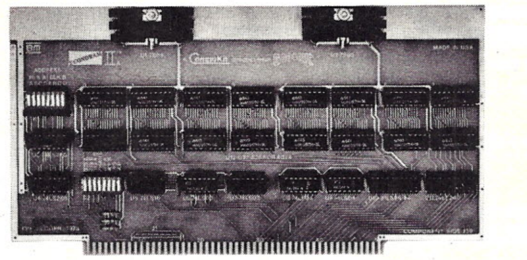
You want speed? Except for the Heath H8 board, *every* Econoram* runs trouble-free with 4 MHz systems. You want low power? We use low power static parts exclusively, which means our *maximum* guaranteed current specs are often lower than the *typical* current specs of our competitors. You want low cost? Our volume purchasing and technical know-how keeps our products price-competitive with boards that do a whole lot less. You want it now? Not only do we stock what we advertise, but many computer stores both nationwide *and* overseas stock much or all of the CompuKit™ line.

You want a choice of configurations? For lowest price buy an "unkit", with sockets and bypass caps pre-soldered in place for easy assembly; specify assembled and tested boards when you can't wait to get up and running; and if you're dealing with critical applications—like accounting or medical applications—check into the boards qualified under our high-reliability Certified System Component(CSC) program. These boards are burned in for 200 hours minimum and are immediately replaced in the event of failure within 1 year of invoice date.

You want reliable, cost-effective operation? That's how we got our reputation... ask any Econoram owner.

FEATURED BOARD OF THE MONTH:
ECONORAM IIA $149 UNKIT, $164 ASSM

The **8K X 8 Econoram II** has been the workhorse of many an S-100 computer system... and now it's improved, with features like new 4K static chips, guaranteed 4 MHz operation, simpler layout, and low power (draws 8W *maximum* from the 8V buss —and that's a guaranteed spec). We've retained all the popular features of the original, such as deselect switch for 4K operation, dual 4K block configuration with independent addressing, and switchable memory protect for each block. **The oldest board on the block is now one of the newest, and we think you're really going to like it.**



OTHER ECONORAM* FULLY *STATIC* MEMORY BOARDS:

Name	Storage	Buss	Speed	Configuration	Unkit	Assm	CSC
Econoram IV	16K X 8	S-100	4 MHz	1-16K block	**$295**	**$329**	**$429**
Econoram VI	12K X 8	H8	2 MHz	1-8K, 1-4K	**$200**	**$270**	n/a
Econoram VII	24K X 8	S-100	4 MHz	2-4K, 2-8K	**$445**	**$485**	**$605**
Econoram IX	32K X 8	Dig Grp	4 MHz	2-4K, 1-8K, 1-16K	**$649**	n/a	n/a
Econoram X	32K X 8	S-100	4 MHz	2-8K, 1-16K	**$599**	**$649**	**$789**
Econoram XI	32K X 8	SBC	4 MHz	2-8K, 1-16K	n/a	n/a	**$1050**

BANK SELECT MEMORIES (for Alpha Micro Systems, Marinchip, etc.)

Name	Storage	Buss	Speed	Configuration	Unkit	Assm	CSC
Econoram XII-16	16K X 8	S-100	4 MHz	2 indep. banks**	**$369**	**$419**	**$519**
Econoram XII-24	24K X 8	S-100	4 MHz	2 indep. banks**	**$479**	**$539**	**$649**
Econoram XIII	32K X 8	S-100	4 MHz	2 indep. banks**	**$629**	**$699**	**$849**

*__Econoram__ is a trademark of **Bill Godbout Electronics.**

**Econoram XII-16 and -24 have 2 independent banks addressable on 8K boundaries; Econoram XIII has 2 independent banks addressable on 16K boundaries.

OTHER PRODUCTS:

NEW! 2708 EROM BOARD KIT... at a special introductory price.

4 independently addressable 4K blocks, with selective disable for each block. Built to Econoram standards (dipswitch addressing, top quality board, sockets wave-soldered in place), and includes dipswitch selectable jump start built right into the board. Includes all support chips and manual, but does ***not*** include EROMs. Special introductory price through August 1, 1979: **$69.95.** After that, the price goes up to the normal $85... don't say we didn't warn you.

POPULAR ICs AT POPULAR PRICES

Low power **2102s** for 2 MHz systems on special: **10/$9.90. 1791 MOS LSI** dual density disc controller from Western Digital: **$59** with pinout and data. **1771** single density controller: **$22.50.** All parts are offered on a while-they-last basis.

NEW MULLEN COMPUTER PRODUCTS: H8 EXTENDER BOARD KIT $39

We're happy to distribute a kit that really takes the hassle out of troubleshooting or testing the popular H8 computer. Includes jumper links in the power supply lines for insertion of fuses, Ammeters, current limiters, etc.

ACTIVE TERMINATOR KIT $34.50

As written up in the April '79 issue of ***Kilobaud Microcomputing.*** Our much imitated design plugs into any S-100 motherboard to reduce ringing, crosstalk, noise, and other buss-related problems.

S-100 MOTHERBOARDS

11 slot unkit: $90. 18 slot unkit: $124. Each motherboard includes all edge connectors wave-soldered in place for easy assembly, integral active termination circuitry, extra wide power and ground traces, and much more.

TERMS: Allow 5% shipping, excess refunded. Cal res add tax. VISA®/Mastercharge® call our 24 hour order desk at **(415) 562-0636.** COD OK with street address for UPS (COD charge applies). Prices good through cover month of magazine. Orders under $15 add $1 handling.

CompuPro™ from GODBOUT ELECTRONICS

G4

FREE FLYER: We'll be glad to tell you more than the space of this ad permits. Just send your name and address, we'll take care of the rest. If you're in a hurry, enclose 41¢ in stamps for 1st class delivery.

DEALER DIRECTORY

Shop the store nearest you.

Peoria AZ

Discount prices: North Star, Pertec disk drives, Integrand mainframes, Godbout memories, Tarbell disk controllers, TEI mainframes, Verbatim diskettes, custom hardware/software, service, repairs & consulting. **Thought Works, 10425 W. Seldon Ln., Peoria AZ 85345, 972-4065.**

Los Angeles CA

Featuring: PolyMorphic, North Star, Imsai, Cromemco, Extensys, Speechlab products and Poly-88 Users Group software exchange. All products 10-20% off list. We won't be undersold! **A-A-A-A Discount Computer How's, 1477 Barrington, Suite 17, Los Angeles CA 90025, 477-8478.**

Palo Alto CA

Systems available for immediate delivery: word processing; multiprogramming, multiuser work in process; business; medical/dental billing and accounts receivable. Software and hardware guaranteed. **Byte of Palo Alto, 2233 El Camino Real, Palo Alto CA 94306, 327-8080.**

San Leandro CA

From personal computers to small systems: North Star, Horizon, DD, Sorcerer, Godbout, TRS-80, Imsai, Verbatim, terminals, printers, floppies, boards, software, books, magazines. Custom programming. **Computer Store of San Leandro, 701 MacArthur Blvd., San Leandro CA 94577, 569-4174.**

Torrance CA

If you're serious, come see us! Complete systems for business, special and personal applications. Packaged and proprietary software. Custom programming. Quality: Apple II, Exidy, Alpha Micro and CP/M systems. Serving southern California better. **Omega Micro Computers, 3535 Torrance Blvd., Suite 10, Torrance CA 90503, 370-9456.**

Denver CO

Experimenters' Paradise. Electronic and mechanical components. Computer People, Audio People, Hams, Robot Builders, Experimenters. Open six days a week. **Gateway Electronics Corp., 2839 W. 44th Ave., Denver CO 80211, 458-5444.**

Wilmington DE

Turnkey biz system. Hardware and software. Custom formatting available; checks, invoices, reports. Micropolis drives/MDOS/BASIC. FORTRAN/CP-M/CBASIC possible. SASE for our low prices. **ComputerEASE, Inc., PO Box 4156, Wilmington DE 19807, 998-5589.**

Pompano Beach FL

Business systems, personal systems, whatever the application, we can help. Consulting, programming, education and maintenance. Service, support and professionalism at affordable prices. **Computer Age Inc., 1308 N. Federal Hwy., Pompano Beach FL 33062, 946-4999.**

Arlington Hts. IL

Lowest prices, never undersold, postpaid in USA—Teletype 43 keyboard printers—Midwest Scientific Instruments computers—SS-50 bus peripherals—open 6-11 PM daily. **Data Mart, 914 East Waverly Drive, Arlington Heights IL 60004, 398-8525.**

Chicago IL

Staffed by computer professionals. Largest selection of books and magazines in the Chicago area. Specialists in CP/M, Apple Computer and Alpha microsystems. **Data Domain of Schaumburg, Plaza de las Flores, 1612 E. Algonquin Rd., Schaumburg IL 60195, 397-8700.**

Naperville IL

Discount Prices: Creed Teletypes, optical scanners, Solid State Music products, Integrand mainframes, Vista floppy-disk systems, many other items, complete hardware and software systems. **Wilcox Enterprises, 25W178-39th St., Naperville IL 60540, 420-8601.**

Laurel MD

Exidy Sorcerer & accessories, Central Data memory boards, G2 software library, full line of ham & SWL equipment. **The Comm Center, Inc., Laurel Plaza, Rte. 198, Laurel MD 20810, 792-0600. Washington: 953-9535.**

Boston MA

Largest computer store in New England with widest selection of microprocessor and LSI 11-based systems. Two miles north of the Mass. Pike, just off Route 128, exit 49. **Computer Mart of Massachusetts, 1395 Main Street (Rt. 117), Waltham MA 02154, 899-4540.**

Westboro MA

Supplies and accessories for the TRS-80. Minidisks; printer paper and forms. Colorful fabric dust covers. Line filters, programs. Custom programming and consulting. Computerized bookkeeping service. **August Automation, 28 Milk St., Westboro MA, 366-0870.**

Worcester MA

Computer products for personal and business systems. Stocking full line of Instant Software, G2 and Personal Software for TRS-80, Apple, PET, Sorcerer. Open six days. **Computer Packages Unlimited, Centerwood Terrace, 244 West Boylston St., West Boylston MA 01583, 829-2570.**

Grand Rapids MI

Full-line microcomputer store. Ohio Scientific—Equinox—PolyMorphic Systems—Digital Systems—Godbout—Dynabyte—Thinker Toys—Meca—North Star. **Micro Computer World, 313 Michigan St., N.E., Grand Rapids MI 49503, 451-8972.**

St. Paul MN

Ohio Scientific Authorized Dealer. Complete business and personal computer systems. Store hours: Mon-Sat 9-5. Custom software design and programming. To take advantage of our many years of computer experience call, write or visit: **Frisch Computer Systems, Inc., 1415 Arcade St., St. Paul MN 55106, 771-7569.**

St. Louis MO

Experimenters' Paradise. Electronic and mechanical components. Computer People, Audio People, Hams, Robot Builders, Experimenters. Open six days a week. **Gateway Electronics Corp., 8123-25 Page Blvd., St. Louis MO 63130, 427-6116.**

Oilmont MT

Business Systems for accounting and word processing. Business software for 6800 MSI, Mini-Flex and North Star CBASIC. Our products save time and money. **Compumont, Inc., 2708 W. Main, Oilmont MT 59466, 337-3015.**

Rome-Utica NY

Ohio Scientific Authorized Dealer. Computers for personal, business, education and industry. Software for accounting, information management, word processing, education, business, personal and games. **Comp-u-Ware, 9057 Main St., Westernville NY 13486, 827-4487.**

White Plains NY

Processor Technology, North Star HORIZON, Apple II, Commodore PET and KIM; full line of books, components and peripherals; classes in our Seminar Room. **The Computer Corner, 200 Hamilton Avenue, White Plains NY 10601, 949-3282.**

Akron OH

We've got it all. Business systems. Personal systems. Software packages. Custom programming. Terminals. Printers. Service and books. Easy freeway access. 10 AM to 6 PM Monday-Saturday. **The Basic Computer Shop, Fairlawn Plaza, 2671 West Market St., Akron OH 44313, 867-0808.**

York PA

Personal and small-business systems. Components and peripherals. Books, magazines, information. Hobbyists welcome. 1/2 block from Business 83 at Iron Bridge. **G.Y.C. Co., 51 Hamilton Ave., York PA 17404, 854-0481.**

Houston TX

Experimenters' Paradise! Electronic and mechanical components for computer people, audio people, hams, robot builders, experimenters. Open six days a week. **Gateway Electronics, Inc., 8932 Clarkcrest, Houston TX 77063, 978-6575.**

Houston TX

Bare boards, kits, small systems, multiuser systems, custom hardware design, software for business or hobby, nationwide consulting available. Please call. **Houston Computer Mart, 8029 Gulf Freeway, Houston TX 77017, 649-4188.**

Houston TX

TRS-80 professionals! Business systems for immediate delivery. Complete business and hobby software library. Custom programming. Centronics, TI, Pertec, NEC, Selectraprint. Modifications and repair of TRS-80. **Houston Micro-Computer Technologies, Inc., 5313 Bissonnet, Bellaire TX 77401, 661-2005.**

Richland WA

Apple, North Star, Ohio Scientific. All products listed in Byte Shopper, classes, literature, electronic parts, repair service. Custom design/consulting in both hardware and software. **Ye Olde Computer Shoppe, inc., 1301 George Washington Way, Richland WA 99352, 946-3330.**

Madison WI

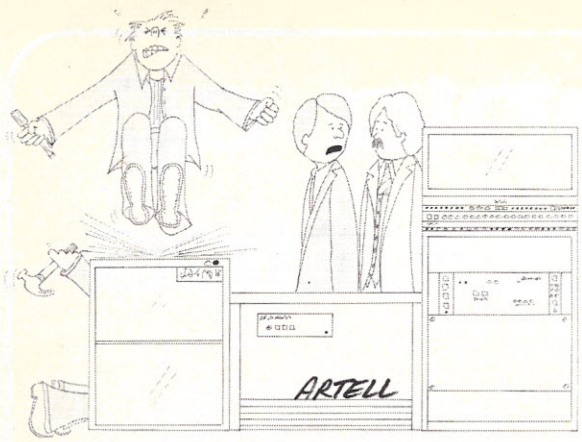
Specializing in complete tested systems for unique applications. Cromemco, TEI, Compucolor II, Godbout, North Star, Vector Graphic, Hazeltine, Perkin-Elmer, U.S. Robotics and much more. **Magic Lantern Computers, 6414 Copps Ave., Madison WI 53716, 221-8643.**

Dealers: Listings are **$15** per month in prepaid quarterly payments, or one yearly payment of **$150**, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-7138 or write *Kilobaud MICROCOMPUTING*, Ad Department, Peterborough NH 03458.

CORRECTIONS

Our printer inadvertently omitted the by-lines for "What's so Magic about the Sorcerer?" (p. 100) and "Personal Computing, Meet Photography" (p. 114) in the June 1979 issue. For anyone wishing to correspond with authors Ken Barbier or Michael Avery, their respective addresses are: Borrego Engineering, PO Box 1253, Borrego Springs CA 92004, and 3200 Larry Lane, Austin TX 78702.

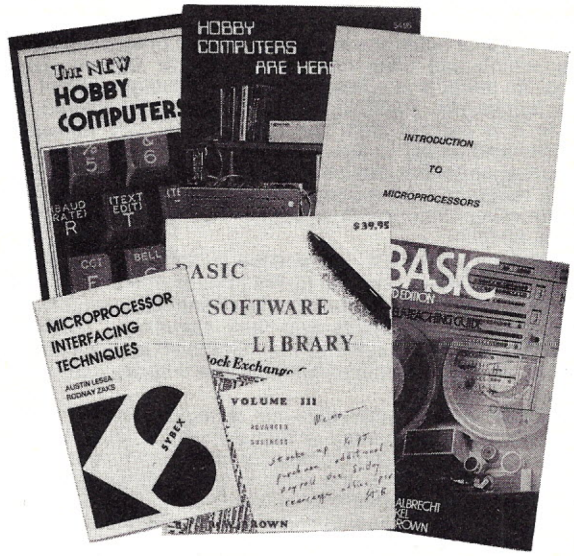
Ray Boaz, author of "Super Standard/High-Speed Cassette Interface" (May 1979, p. 42), sent us a few corrections to his article. The first is in Table 2. Power wiring—the voltage is shown as −5 V; it should be +5 V. The second is in Fig. 3. The line connecting U2a-1, U2b-13 and U3a-1 should also have been connected to the other pins held high by the "pull up" gate U4c-8. The text indicated that 312 microseconds is 75 percent of a 2400 Hz period. It should have stated "approximately 75 percent"; however, anyone within ±0.5 microsecond may be assured that is close enough.



"Believe it or not, they're the best maintenance team in the region."

"Your headphones are plugged into your computer."

kb microcomputing book nook



●**HOBBY COMPUTERS ARE HERE!**—BK7322—If you want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

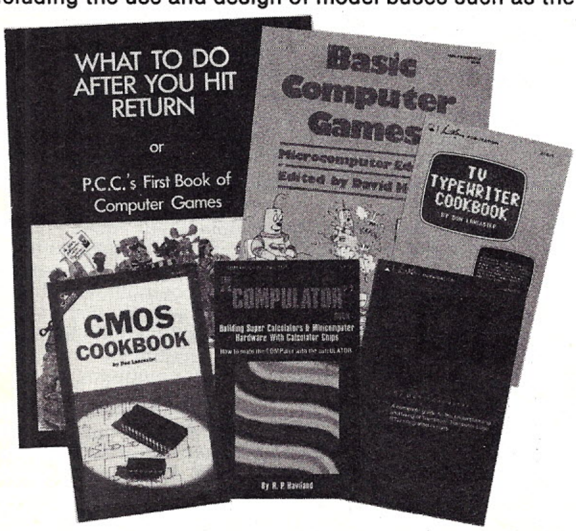
●**THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where **Hobby Computers Are Here!** leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! $4.95*

●**INTRODUCTION TO MICROPROCESSORS**—BK1032—by Charles Rockwell of MICROLOG, is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. Describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Specific systems are not described and programming is only briefly discussed. $17.50 U.S. and Canada; $20 elsewhere.*

● **BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $4.95.*

● **SCIENTIFIC RESEARCH INSTRUMENT'S BASIC SOFTWARE LIBRARY** is a complete do-it-yourself kit, written in everybody's BASIC, immediately executable in ANY computer with 8K; no other peripherals needed. **Volume I** contains business and recreational programs and is 300 pages. **Volume II** is 260 pages and contains math, engineering, statistics and plotting programs. **Volume III** contains money managing, advanced business programs such as billing, A/R, inventory, payroll, etc. **Volume IV** contains general purpose programs such as loans, rates, retirement, plus games: Poker, Enterprise (take charge while Capt. Kirk is away), Football and more! **Volume V** is filled with experimenters' programs, including games, pictures and misc. problems such as "logic." Volume I—LB1002—& Volume II—LB1003—$24.95* each, Volume III—LB1004—$39.95*, Volume IV—LB1005—& Volume V—LB1006—$9.95* each.

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $9.95.*



●**TTL COOKBOOK**—BK1063—by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $8.95.

●**CMOS COOKBOOK**—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $9.95.*

●**TVT COOKBOOK**—BK1064—by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*

●**THE "COMPULATOR" BOOK—BUILDING SUPER CALCULATORS & MINICOMPUTER HARDWARE WITH CALCULATOR CHIPS**—BK1012—by R.P. Haviland, provides ideas, design info and/or printed circuit boards for calculator chip projects such as tape control of calculator electronic lock, dial-a-telephone, etc. $7.95.*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● **WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing's* dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is **Assembly/Machine Language** (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for **BASIC** (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud **Microcomputing** Book Department ● **Peterborough NH 03458. Be sure** to include check or detailed credit card information.
All orders, add $1.00 handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $9.95.*

● **MICROPROCESSOR LEXICON—ACRONYMS AND DEFINITIONS**—BK1040—compiled by the staff of SYBEX, is a convenient reference in pocket-size format. Sections include acronyms and definitions, part numbers and their definitions, S-100 signals; RS232 signals, IEEE 499 signals, microcomputers and microprocessors, JETDS summary (military) and a code conversion table. $2.95.*

● **INSTANT BASIC**—BK1127—by Jerald R. Brown. For the personal computer enthusiast or the user of DEC's BASIC PLUS language, here is a new book to teach you BASIC. It teaches BASIC to beginners using interesting programming ideas and applications that will be easily understood by the home computer programmer. BASIC PLUS users know that the two languages are very similar, so this book can be used by them as well. This is an "active participation" workbook, designed to be used with your home computer so you can learn by doing! Ideas are slowly introduced in a nonmathematical context so the beginner can quickly learn good programming techniques. $6.00.*

● **MY COMPUTER LIKES ME . . . WHEN I SPEAK BASIC**—BK1039—An introduction to BASIC . . . simple enough for kids. If you want to teach BASIC to anyone quickly, this is the way to go. $2.00.*

● **COMPUTER PROGRAMMING HANDBOOK**—BK1014—by Peter Stark. A complete guide to computer programming and data processing. Includes many worked-out examples and history of computers. $9.95.*

● **MICROCOMPUTER DICTIONARY**—BK1034—This microcomputer dictionary fills the need to become quickly acquainted with the terminology and nomenclature of the revolution in computers. There is also a comprehensive electronics/computer abbreviations and acronyms section. $15.95.*

● **YOUR HOME COMPUTER**—BK1172—by James White, is an introduction to the world of personal microcomputing. This book tells you everything you want to know about home computing and gives the computer novice a painless introduction to microcomputer technology and terminology, beginning with what computers are and how they work. This basic book requires no prior knowledge or experience in electronics or computing. It provides information about home computer kits; guidelines for selecting and building your own microcomputer, how to use your home computer and what you can do with it; lists of computer stores, clubs, periodicals; and answers to many more of your questions about microcomputers and the jargon surrounding the personal computing scene today. $6.00.*

● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

THE BASIC HANDBOOK—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

● **SCELBI'S FIRST BOOK OF COMPUTER GAMES**—BK1048—Need a game for your 8008 or 8080 microprocessor? This book has three popular games: "Space Capture," "Hexpawn" and "Hangman." Complete flowcharts, logic description, program listing and instructions are provided. A must for the game freak! $14.95.*

● **MICROCOMPUTER PRIMER**—BK1035—by M. Waite and M. Pardee. Describes basic computer theory, explains numbering systems and introduces the reader to the world of programming. Describes the world of microcomputing in "real world" terminology. $7.95.*

● **THE STORY OF COMPUTERS**—BK1056—by Donald D. Spencer, is to computer books what *Dick and Jane* is to novels . . . elementary, gives the non-computerist a fair idea of what the hobbyist is talking about when he speaks computer lingo. Attempts to explain what computers are and can do to a spouse, child or any non-electronics-minded friend. $4.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud **Microcomputing** Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information.
All orders, add $1.00 handling.

FOR CUSTOMER SERVICE CALL 603-924-7298

kb microcomputing book nook

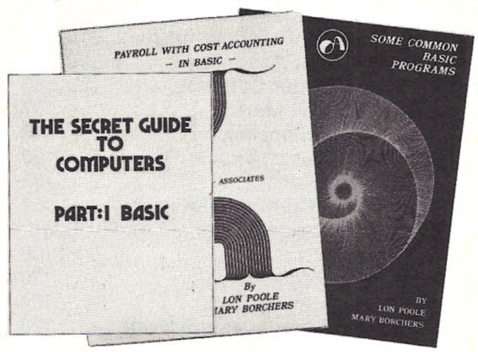
● AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*

● VOL. I—BK1030—Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $9.50.*

● VOL. II—BK1031 (loose leaf)—$25.00*; BK1040 (with binder)—$30.00*—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips.

● VOL. III—BK1132 (loose leaf)—$15.00*; BK1133 (with binder)—$20.00.* Contains descriptions of all support devices that can be used with any microprocessor.

● FUN WITH COMPUTERS AND BASIC—BK1021—by Donald D. Spencer, contains an easy-to-understand explanation of the BASIC Programming Language and is intended for persons who have had no previous exposure to computer programming. Over half the book is devoted to problems using games, puzzles, and mathematical recreations. A superior book for self-teaching and learning computer programming. $6.95.*



● ADVANCED BASIC—BK1000—Applications and problems by James Coan is for those who want to extend their expertise with BASIC. Offers advanced techniques and applications. $7.95.*

● PAYROLL WITH COST ACCOUNTING—IN BASIC—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941. $12.50.*

● SOME COMMON BASIC PROGRAMS—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $9.50 paperback.*

● THE SECRET GUIDE TO COMPUTERS Parts 1, 2 and 3 by Russ Walter. **Part One** describes computers in general, and after reading for ten minutes you will be writing simple BASIC programs! **Part Two** discusses computer applications. It's one thing to master the syntax of the language such as BASIC and another to solve problems using the new tool. **Part Three** describes programming languages. Ever heard of APL and QLISP? BASIC is not the only language used to program computers. 7th Edition. Part I—BK1050—$2.75*; Part II—BK1051—$2.50*; Part III—BK1052—$3.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud **Microcomputing** Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information.
All orders, add $1.00 handling.

KB BOOK NOOK ORDER FORM

Qty.	Cat. #	Description	Unit Price	Total
			Add $1 Handling	
			Total	

Enclosed $ ________ ☐ Check ☐ Money Order

☐ Master Charge ☐ VISA ☐ American Express

Credit card #________________ Exp. date________ Signature________________

Name________________________

Address________________________

City____________ State________ Zip________

Coupon expires in 60 days

kilobaud
MICROCOMPUTING T.M.
Peterborough NH 03458

KB 7/79

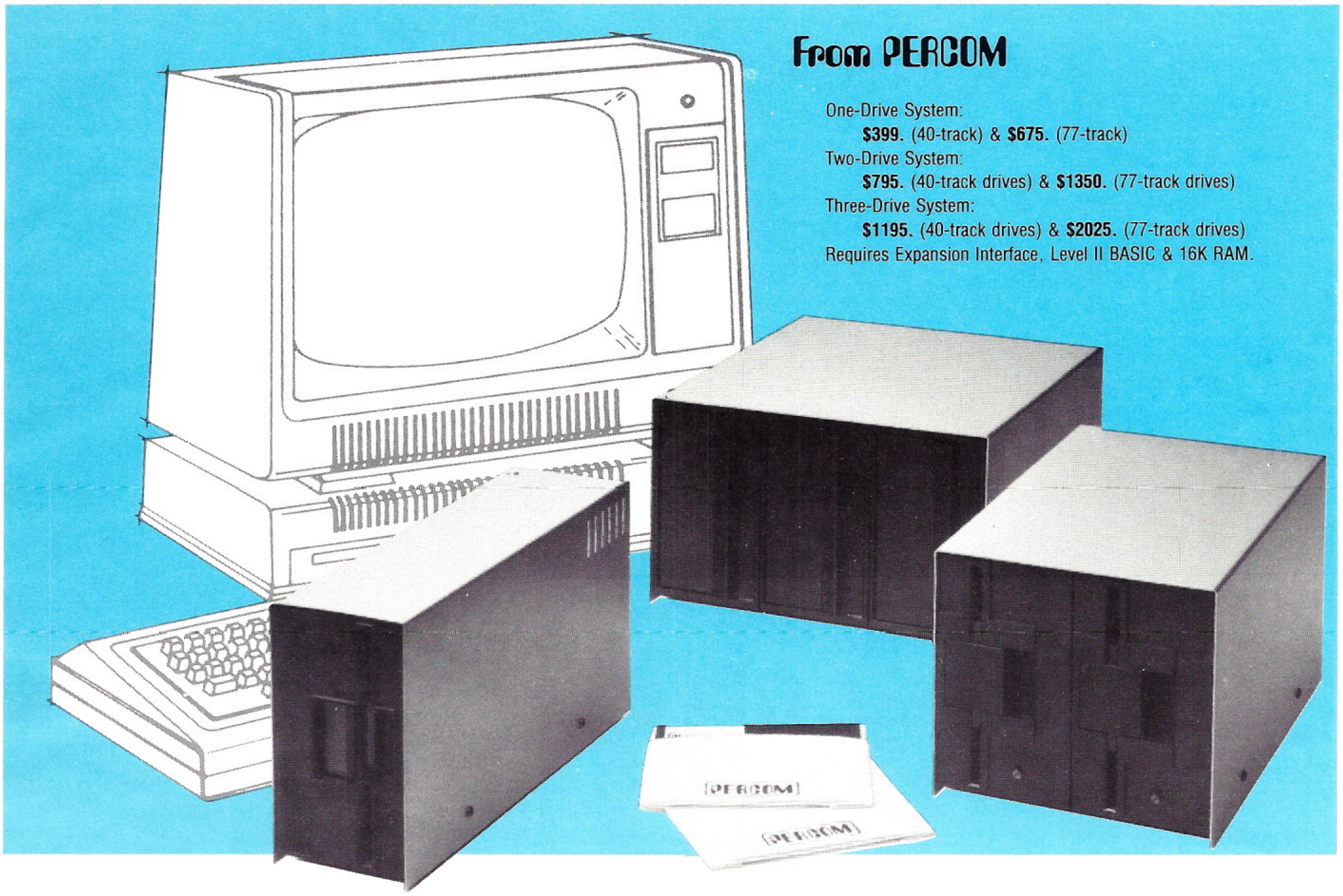
FOR TOLL FREE ORDERING CALL 1-800-258-5473

kilobaud

MICROCOMPUTING T.M.

LIST OF ADVERTISERS

*Reader Service inquiries not honored. Please contact advertiser directly.



Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get more for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!

PERCOM

✓P7

PERCOM DATA COMPANY, INC.
DEPT. K • 211 N. KIRBY • GARLAND, TX. 75042

To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

Percom *'peripherals for personal computing'*

Small Computers For Big Jobs

Personal Computer Place
Mesa, AZ

Phoenix Micro Systems
Mesa, AZ

A-Vidd Electronics
Long Beach, CA

Software Dynamics
Anaheim, CA

Datek
Newington, CT

Fuel Computer Corp.
New Haven, CT

Harper Buffing
E. Hartford, CT

Microcomputer Systems
Tampa, FL

American Microprocessors
Half Day, IL

Computerland
Arlington Heights, IL

Data Mart
Arlington Heights, IL

Great Lakes Industries
Elgin, IL

Lillipute Computer Mart
Skokie, IL

Midland Standard
Elgin, IL

Electronic Enterprises
Logansport, IN

Johnson Controls
Goshen, IN

Electronic Data
Des Moines, IA

Barney & Assoc.
Pittsburg, KS

CMPTR-C
Topeka, KS

Micro Dynamics
Olathe, KS

Freeman Micro Tech.
W. Monroe, LA

Crockergraphics
Needham Heights, MA

Pacer Systems
Burlington, MA

O-M Sales
Ann Arbor, MI

Gallion Data System
Rockaway Beach, MO

H & K Computer Corp.
Kansas City, MO

Compact Computers
Butte, MT

Acutronics
Summit, NJ

Beacon Computer
Metuchen, NJ

High Technology
Oklahoma City, OK

The Electronics Place
Pittsburgh, PA

Gallion Data System
Camp Hill, PA

G.Y.C. Company
York, PA

International Data
Solebury, PA

The Computer Shop
San Antonio, TX

Barbara Rush
Mesquite, TX

Small Computer Products
Austin, TX

Southwest Electronics
Melbourn, Australia

Sontron Instruments
Victoria, Australia

Arelco
Belgium

The Computer Circuit
Ontario, Canada

Strumech Engineering
Staffordshire, England

MRL Building
Delft, Holland

Hauker HF
Reykjavid, Iceland

Electricity Supply Board
Ireland

Digito Systemas S.A.
Monterrey, Mexico

ECONZ
New Zealand

Radiokom
Transvaal, South Africa

Testas Turkish Electronics
Ankara, Turkey

Sistema 6.600
Caracas, Venezuela

C.O.I. - Systeme Munchen
West Germany

MSI Midwest Scientific Instruments

220 W. Cedar • Olathe, KS 66061 • (913) 764-3273 • TWX 910 749 6403 (MSI OLAT) • TELEX 42525 (MSI OLAT)